



DEPARTAMENTO DE MATEMÁTICA E ENGENHARIAS

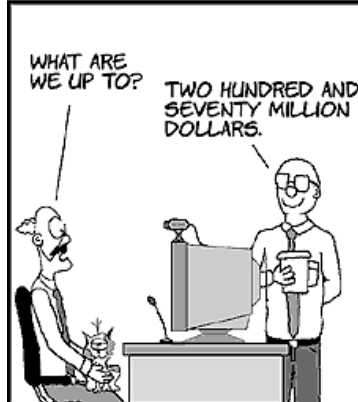
# **OS ROBOTS NAS AULAS DE INFORMÁTICA PLATAFORMAS E PROBLEMAS**

**Luís Alberto da Silva Gaspar**  
(Licenciado)

*Tese Submetida à Universidade da Madeira para a  
Obtenção do Grau de Mestre em Engenharia Informática*

Funchal – Portugal  
Dezembro 2007

USER FRIENDLY by Illiad



COPYRIGHT © 2001 ILLIAD HTTP://WWW.USERFRIENDLY.ORG/



Fonte: <http://ars.userfriendly.org/cartoons/?id=20010723>

**Orientador:**

**Professor Doutor Eduardo Leopoldo Fermé**

*Prof. Associado no Departamento de Matemática e Engenharia da Universidade de Madeira*

---

# Resumo

---

Desde há alguns anos que vêm sendo desenvolvidas, em vários sítios do mundo, experiências com a utilização de Robots como uma ferramenta educativa, com especial incidência ao nível do Ensino Superior e em alguns casos no Ensino Básico ou Secundário.

Neste trabalho de investigação pretende-se fazer uma análise crítica sobre o uso dos robots no ensino da programação nas disciplinas de Informática do Ensino Secundário e disciplina de Inteligência Artificial da Licenciatura do Curso de Engenharia Informática.

Com o objectivo de usar o robot como mediador entre o aluno e o ensino da programação, identificamos os conteúdos programáticos das diversas disciplinas do Ensino Secundário e fizemos o levantamento de ferramentas e soluções tecnológicas existentes que pudessem ser aplicadas nas aulas.

Como contributo desta investigação pretende-se: (i) disponibilizar uma série de problemas adequados aos vários conteúdos programáticos, para serem utilizados nas salas de aulas; (ii) criar e otimizar ferramentas, mais concretamente plataformas de programação, para os alunos resolverem os problemas através dos robots.

---

# Abstract

---

For some years that have been developed in several sites in the world, experiments with the use of Robots as an educational tool, with particular reference to the level of Higher Education and in some cases in the primary or secondary education.

This research work aims to make a critical analysis on the use of robots in the teaching of programming in the disciplines of Informatics of Secondary and discipline of Artificial Intelligence's Degree Course of Engineering Informatics.

In order to use the robot as a mediator between the student and teaching of programming, identified the programmatic content of the various disciplines of Secondary and made the lifting of existing tools and technology solutions that could be applied in the classroom.

As a contribution of this research plan is to: (i) provide a series of problems appropriate to the various programmatic content to be used in rooms of classes, (ii) creating and optimizing tools, specifically platforms programming, for students solve problems through the robots.

---

# Palavras Chave

---

Ensino e Aprendizagem, Robótica Educativa, Lego® Mindstorm™, Programação, Inteligência Artificial, Plataformas de Programação, Problemas.

---

# Keywords

---

Teaching and Learning, Educational Robotics, Lego® Mindstorm™, Programming, Artificial Intelligence, Programming Platforms, Problems.

*Aos meus Pais e Irmão*  
*Em especial à Andreína*

---

# Agradecimentos

---

O meu agradecimento é dirigido ao Professor Doutor Eduardo Leopoldo Fermé, não só pela orientação deste trabalho, como pela confiança, entusiasmo, disponibilidade, paciência e amizade que sempre me dispensou. Proporcionou ainda as condições necessárias para a elaboração desta dissertação, com apoio material e logístico através do projecto DROIDE e da disciplina de Inteligência Artificial, que permitiram a minha integração e evolução do trabalho aqui apresentado.

À Andreína, por todo o seu amor e apoio.

À minha família, em especial a minha mãe e irmão, por todo o apoio incondicional que sempre me deram.

À Dolores e Nelson, pela amizade e ajuda na revisão da dissertação.

Aos elementos do projecto DROIDE, pelo apoio material e logístico.

Aos meus amigos, Elci Alcione, Patrícia Silva, Paulo Rodrigues, Vítor Pereira, Rui Alves, Carlos Silva, Jordão Abreu, Feliz Pereira, Pascal e Élvio Jesus que foram apoiando e perguntando pelo trabalho e suportaram as minhas ausências.



---

# Índice

---

RESUMO .....	III
ABSTRACT .....	IV
PALAVRAS CHAVE .....	V
KEYWORDS .....	V
AGRADECIMENTOS .....	VII
ÍNDICE .....	VIII
ÍNDICE DE FIGURAS .....	X
ÍNDICE DE TABELAS .....	XI
ACRÓNIMOS .....	XII
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1. PORQUÊ OS ROBOTS LEGO® MINDSTORMS™ .....	2
1.2. MOTIVAÇÃO E OBJECTIVOS .....	3
1.3. OS ROBOTS COMO MEDIADORES ENTRE O ALUNO E A INFORMÁTICA .....	4
1.4. CONTRIBUIÇÕES DA DISSERTAÇÃO .....	5
1.5. ORGANIZAÇÃO DA DISSERTAÇÃO .....	6
<b>2. ESTADO DA ARTE .....</b>	<b>9</b>
2.1. ROBÓTICA EDUCATIVA .....	9
2.2. A UTILIZAÇÃO DOS ROBOTS DA LEGO® MINDSTORMS™ NO ENSINO DA INFORMÁTICA .....	11
2.2.1. Competições .....	11
2.2.2. Projectos e Iniciativas .....	12
2.3. LIMITAÇÕES E DESAFIOS .....	16
2.4. CONCLUSÃO .....	16
<b>3. DESCRIÇÃO TEÓRICA DOS ROBOTS .....</b>	<b>19</b>
3.1. O KIT LEGO® MINDSTORMS™ .....	20
3.1.1. O RCX .....	20
3.1.2. Os Motores .....	23
3.1.3. Os Sensores .....	23
3.1.4. A Torre de Transmissão .....	26
3.2. O FIRMWARE .....	26
3.2.1. LEGO .....	27
3.2.2. BrickOS .....	27
3.2.3. Lejos .....	27
3.3. FERRAMENTAS DE PROGRAMAÇÃO EXISTENTES .....	28
3.3.1. Robotics Invention System .....	28
3.3.2. ROBOlab™ .....	29
3.3.3. Bricx Commander Center .....	30
3.4. LINGUAGENS DE PROGRAMAÇÃO .....	31
3.4.1. Linguagem Nativa – NQC .....	32
3.4.2. Linguagens não Nativas .....	33
3.5. CONCLUSÃO .....	34
<b>4. ENSINO DA INFORMÁTICA EM PORTUGAL .....</b>	<b>37</b>
4.1. A FINALIDADE DO ENSINO DE INFORMÁTICA .....	37
4.2. ANÁLISE CURRICULAR DOS PROGRAMAS .....	38
4.2.1. Aplicações Informáticas A - 10.º ano .....	38

4.2.2.	<i>Aplicações Informáticas A - 11.º ano</i>	40
4.2.3.	<i>Bases de Programação 10.º, 11.º e 12.º anos</i>	41
4.2.4.	<i>Tecnologias Informáticas 10.º, 11.º e 12.º anos</i>	42
4.2.5.	<i>Técnicas de Gestão de Bases de Dados 12º Ano</i>	44
4.3.	ONDE APLICAR OS ROBOTS	45
4.4.	CONCLUSÃO	46
<b>5.</b>	<b>NOVAS PLATAFORMAS DE PROGRAMAÇÃO</b>	<b>47</b>
5.1.	RCX+PROLOG	47
5.1.1.	<i>Motivação para a construção da plataforma</i>	48
5.1.2.	<i>Funcionamento da Plataforma</i>	51
5.1.3.	<i>Módulos da Plataforma</i>	53
5.2.	RCX+PASCAL	55
5.2.1.	<i>Motivação para a construção da plataforma</i>	55
5.2.2.	<i>Funcionamento da Plataforma</i>	56
5.2.3.	<i>Módulos da Plataforma</i>	57
5.3.	CONCLUSÃO	59
<b>6.</b>	<b>PROBLEMAS</b>	<b>61</b>
6.1.	ACTIVIDADE 1 - INTRODUÇÃO À LINGUAGEM PASCAL	65
6.2.	ACTIVIDADE 2 - VARIÁVEIS E EXPRESSÕES	67
6.3.	ACTIVIDADE 3 – ESTRUTURAS DE DECISÃO	69
6.4.	ACTIVIDADE 4 – ESTRUTURAS DE REPETIÇÃO	73
6.5.	ACTIVIDADE 5 - ESTRUTURAS DE REPETIÇÃO + ESTRUTURAS DE DECISÃO	77
6.6.	ACTIVIDADE 6 - VECTORES E MATRIZES	79
6.7.	CONCLUSÃO	83
<b>7.</b>	<b>OS ROBOTS EM ACÇÃO</b>	<b>84</b>
7.1.	ACÇÃO DE FORMAÇÃO	84
7.2.	DISCIPLINA DE INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL	87
7.2.1.	2005/2006 – “ <i>O Mundo de Wumpus</i> ”	88
7.2.2.	2006/2007 – “ <i>O Problema dos Canibais e Missionários</i> ”	91
7.3.	CONCLUSÃO	95
<b>8.</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	<b>96</b>
8.1.	CONCLUSÃO	96
8.2.	TRABALHOS FUTUROS	98
	<b>REFERÊNCIAS</b>	<b>100</b>
	<b>ANEXOS</b>	<b>104</b>
	<b>ANEXO A - MANUAL DA API DE PASCAL</b>	<b>105</b>
	<b>ANEXO B- GUIA DE INSTALAÇÃO E UTILIZAÇÃO DA PLATAFORMA RCX+PROLOG</b>	<b>114</b>
	<b>ANEXO C – PROTOCOLO DE COMUNICAÇÃO DA PLATAFORMA RCX + PROLOG</b>	<b>121</b>
	<b>ANEXO D – PROBLEMA USADO NA ACÇÃO DE FORMAÇÃO</b>	<b>123</b>

---

# Índice de Figuras

---

Figura 1 – Organização da Dissertação .....	6
Figura 5 - O RCX da Lego MindStorms .....	20
Figura 6 – A Estrutura Lógica do RCX.....	22
Figura 7 – Motor.....	23
Figura 8 – Sensor de Luz.....	24
Figura 9 - Sensor de Toque .....	24
Figura 10 - Sensor de Rotação.....	25
Figura 11 - Sensor de Temperatura .....	25
Figura 12 – Torre de Infravermelhos.....	26
Figura 2 - Ambiente de programação do RIS.....	28
Figura 3 - Ambiente de programação do ROBOLAB™.....	30
Figura 4 – Ambiente de programação do BricxCC .....	31
Figura 13 – O RCX como um Agente .....	49
Figura 14 - Arquitectura do RCX+PROLOG.....	51
Figura 15 - Funcionamento do RCX+PROLOG .....	52
Figura 16 - Arquitectura do RCX+Pascal .....	56
Figura 17 – Funcionamento do RCX + PASCAL .....	57
Figura 18 – “O Tanque” .....	62
Figura 19 – “O Todo - Terreno” .....	62
Figura 20 – Proposta de trabalho para o aluno .....	62
Figura 21 - Sugestões metodológicas para o professor .....	63
Figura 22 - Construção do robot.....	85
Figura 23 – Execução do exercício 3.....	86
Figura 24 – “Pára-choques inteligente” .....	86
Figura 25 – “O Mundo de Wumpus” – O original e o modificado .....	89
Figura 26 – Exemplos de robots construídos pelos alunos.....	89
Figura 27 – Alguns robots propostos.....	92
Figura 28 – Ambientes diferentes para resolver o problema.....	92

---

# Índice de Tabelas

---

Tabela 1 - Especificações do Hardware do RCX .....	21
Tabela 2 - Exemplo de um programa escrito na linguagem NQC. ....	32
Tabela 3 - Exemplo de um programa escrito na linguagem Java.....	33
Tabela 4 - Exemplo de um programa escrito na linguagem Pascal.....	34
Tabela 5 – Definições de IA [RN2003] .....	48
Tabela 6 – Estrutura do Protocolo de Comunicação do RCX+PROLOG.....	52
Tabela 7 – Excerto do código Prolog .....	53
Tabela 8 – Excerto do Código NQC.....	54
Tabela 9 – Exemplo de um programa escrito em Pascal .....	58
Tabela 10 – Funções implementadas na biblioteca BIGBLOCKS .....	59
Tabela 11 – Distribuição dos problemas pelos conteúdos programáticos.....	64
Tabela 12 – Transformação de um problema .....	87
Tabela 13 – Excerto de código “O Mundo de Wumpus” .....	90
Tabela 14 - Excerto de código “Os Canibais e Missionários” .....	94

---

# Acrónimos

---

API	Application Programming Interface
BIOS	Basic Input/Output System
IDE	Integrated Development Environment
JVM	Java Virtual Machine
LCD	Liquid Crystal Display
RAM	Random Access Memory
RCX	Robotics Command Explorer
RIS	Robotics Invention System™
ROM	Read-Only Memory
TIC	Tecnologias de Informação e Comunicação



---

# 1. Introdução

---

*“The great thing in this world is not so much where we stand, as in what direction we are moving”*

**Sir Oliver Wendell Holmes**

Hoje os alunos crescem num mundo tecnológico e muitas vezes isso não se reflecte nas escolas. Recorre-se frequentemente às técnicas tradicionais para ensinar.

Qualquer profissional de educação pretende encontrar novas ferramentas que sejam eficazes no processo de ensino/aprendizagem. O objectivo subjacente a esta procura é ter uma ferramenta que motive os alunos e possibilite uma aprendizagem abrangente dos conteúdos. [CCM2006]

Neste contexto, o uso de robots no ensino tem vindo a afirmar-se como uma ferramenta muito útil, em especial na educação científica e tecnológica. No Ensino Básico e Secundário têm sido utilizados maioritariamente em actividades extracurriculares. No Ensino Superior são usados no contexto de projectos de investigação ou em estudos específicos.

A motivação teórica para a utilização de robots na educação vem desde 2000, quando Jonassen argumentou que as ferramentas tecnológicas podem ser vistas como ferramentas cognitivas que aumentam o processo de aprendizagem. Defende ainda que, usando os robots, os alunos descobrem e aprendem as diferentes facetas na resolução de problemas. (citado em [JM2003])

O kit da Lego® Mindstomrs™ tem vindo a se destacar devido à sua flexibilidade a nível de construção e programação do robot, ao seu preço e, não menos importante, pelo

interesse e motivação que proporciona aos alunos. Na secção seguinte apresentamos as razões da escolha deste kit, utilizado no decorrer do trabalho aqui apresentado.

## **1.1. Porquê os robots Lego® Mindstorms™**

---

A escolha dos robots Lego® Mindstorms™ entre os kits de robótica existentes no mercado deve-se essencialmente às seguintes razões [LM]:

### **Mérito e qualidades reconhecidas**

Já foi distinguido com vários prémios. Estudos efectuados apontam boas potencialidades para este produto.

O kit foi desenvolvido no seguimento de outros materiais da Lego, com provas dadas ao nível da sua utilização no ensino.

### **Não exige conhecimentos ao nível da electrónica**

A ausência destes conhecimentos não é impeditiva de o utilizar. A montagem é muito simples, através de encaixes, as ligações eléctricas fazem-se também por encaixe, não exigindo grandes técnicas, como por exemplo soldar, o que poderia vir a ser um entrave à sua utilização.

### **Facilidade de construção e entusiasmo**

Dadas as características deste kit, é muito fácil construir uma determinada estrutura, testar e efectuar as alterações necessárias.

Tem a vantagem de que, salvo raras situações, os alunos já foram ou ainda são admiradores do sistema Lego®, o que proporciona um entusiasmo pela sua utilização.

“(...) estas ferramentas permitem aos alunos construir e testar mais rapidamente (e a um custo mais baixo) do que usando os materiais tradicionais e trazem um grande entusiasmo para o trabalho.” [RP2004]

### **Software produzido**

Programar o robot poderia ser algo eventualmente difícil de transpor. Com o desenvolvimento de software com fins educativos, nomeadamente o RIS e o Robolab, o problema é claramente minimizado. Ainda assim podemos utilizar outras opções de



software ou, até mesmo construir um software próprio, graças à política de “open source” adoptada pela marca.

### **Política adoptada pela marca**

A Lego adoptou uma política de abertura, com o propósito de que interessados e entusiastas aprofundassem o conhecimento do sistema, detectando possíveis falhas, encontrando formas de superar limitações e até produzir novas ferramentas de programação.

### **Preço**

O preço é, naturalmente, um factor a ter em conta. Em comparação com outros robots programáveis o kit da Lego® tem um preço razoavelmente bom, considerando a realidade do sistema educativo e as suas dificuldades orçamentais.

Outra vantagem advém do facto deste kit ser compatível com outros materiais da Lego®, de fácil acesso e do conhecimento dos alunos, havendo a possibilidade de podermos adquirir conjuntos de peças em separado, o que permite complementar o kit de acordo com as necessidades.

## **1.2. Motivação e Objectivos**

---

Desenhar, construir e programar robots, permite aos alunos explorar de uma forma criativa a programação, desenho mecânico, física, matemática, movimento, factores ambientais, resolução de problemas e colaboração em grupo. [DH2000]

Partindo de casos de sucesso como: A aplicação dos Robots Lego® Mindstorms™ para aprender a linguagem C, em que defendem os Robots e a linguagem C juntos, podem ser o ideal para ensinar programação; [HW2005] e do estudo da aplicação dos Robots nas aulas de Inteligência Artificial, em que conclui que os robots promovem uma aprendizagem activa, o que aprofunda os conhecimentos dos alunos. [K2005]

Tendo em conta a falta de trabalhos ou estudos de investigação da aplicação de robots no ensino em Portugal em contextos mais estruturados, como um programa de uma disciplina.

Associando o facto da falta de estudos na preparação de plataformas de programação, de problemas para serem usados em contexto escolar e os casos de sucesso acima apresentados, identificamos os seguintes objectivos para este trabalho de investigação:

- Criar problemas para serem resolvidos, usando os robots, nas disciplinas de programação do ensino secundário;
- Analisar, otimizar e desenvolver plataformas, para serem aplicadas nas disciplinas onde os robots possam ser usados;
- Desenvolver uma plataforma para programar os robots nas aulas de Inteligência Artificial do ensino superior, usando a Linguagem Prolog.
- Criar materiais de suporte para alunos e professores das ferramentas/soluções desenvolvidas.

### **1.3. Os Robots como mediadores entre o Aluno e a Informática**

---

Este trabalho surge como parte integrante de um projecto mais amplo – o projecto “DROIDE: os Robots como Elementos Mediadores entre o Aluno e a Matemática/Informática”, a desenvolver no Departamento de Matemáticas e Engenharias da Universidade da Madeira, [D2005]. Este projecto tem como objectivos:

- Criar problemas na área da Matemática/Informática a serem resolvidos através dos robots;
- Criar robots para abordar problemas específicos na área da Matemática/Informática;
- Implementar a resolução de problemas utilizando robótica nas aulas de matemática no ensino básico e secundário, nas aulas de informática no ensino secundário e nas aulas de Inteligência Artificial, Didáctica da Informática e Didáctica da Matemática no ensino superior;
- Analisar a actividade dos alunos aquando da resolução dos problemas utilizando os robots nos diferentes tipos de aula referidos no ponto anterior.

Este projecto apresenta como resultados parciais as seguintes publicações:

- Crossing the River with Robots: Changing the Way of Working in AI Subject[FF2007];
- RCX+PROLOG: A platform to use Lego Mindstorms(tm) Robots in Artificial Intelligence courses [FG2007];
- Viajando com Robots na Aula de Matemática [FFO2007];
- DROIDE VIRTUAL: UTILIZAÇÃO DE ROBOTS NA APRENDIZAGEM COLABORATIVA DA PROGRAMAÇÃO ATRAVÉS DA WEB [AFF2007];
- UTILIZAÇÃO DE ROBOTS NO ENSINO DE PROGRAMAÇÃO: O PROJECTO DROIDE [FFA2007];
- Using Robots to Learn Functions in Math Class [FFO2006].

## 1.4. Contribuições da Dissertação

---

No sentido de dar resposta aos objectivos propostos, esta dissertação apresenta as seguintes contribuições:

- Realização de um estudo do estado da arte sobre a utilização dos Robots da Lego® Mindstorms™ no Ensino da Informática. Neste estudo é abordada ainda a questão das ferramentas, firmware e linguagens de programação existentes no mercado que podem ser utilizados com este tipo de robots.
- A construção de uma plataforma, designada por nós de RCX + Prolog, para ser usada nas aulas de Inteligência Artificial. Com esta ferramenta pretende-se: (i) Trazer às aulas resultados concretos de forma a motivar os alunos e consequentemente melhorar a sua aprendizagem; (ii) Aliar à utilização dos robots, para a resolução de problemas, a utilização da linguagem de Programação PROLOG.
- Estudo e optimização de uma plataforma, designada por nós de RCX + Pascal, para ser usada nas aulas de bases de programação do 10.º ano. Aqui a preocupação foi criar uma livreria onde criamos funções que se aproximam à filosofia dos blocos de programação (Avançar, Recuar, Virar à direita, entre outros) existentes na plataforma RIS.

- Criação de uma bolsa de problemas, enquadrados nos diversos conteúdos programáticos da disciplina de bases de programação do 10.º ano. Cada um dos problemas está dividido em duas partes, a do aluno e a do professor.
- Elaboração de conteúdos de suporte e ajuda às plataformas de programação que resultaram como contribuição desta dissertação (RCX + PROLOG e RCX+PASCAL).

## 1.5. Organização da Dissertação

---

Esta dissertação encontra-se organizada em oito capítulos, que cobrem as duas áreas em estudo neste trabalho, nomeadamente os robots e o ensino. Na figura seguinte mostramos a distribuição destes capítulos pelas duas áreas.

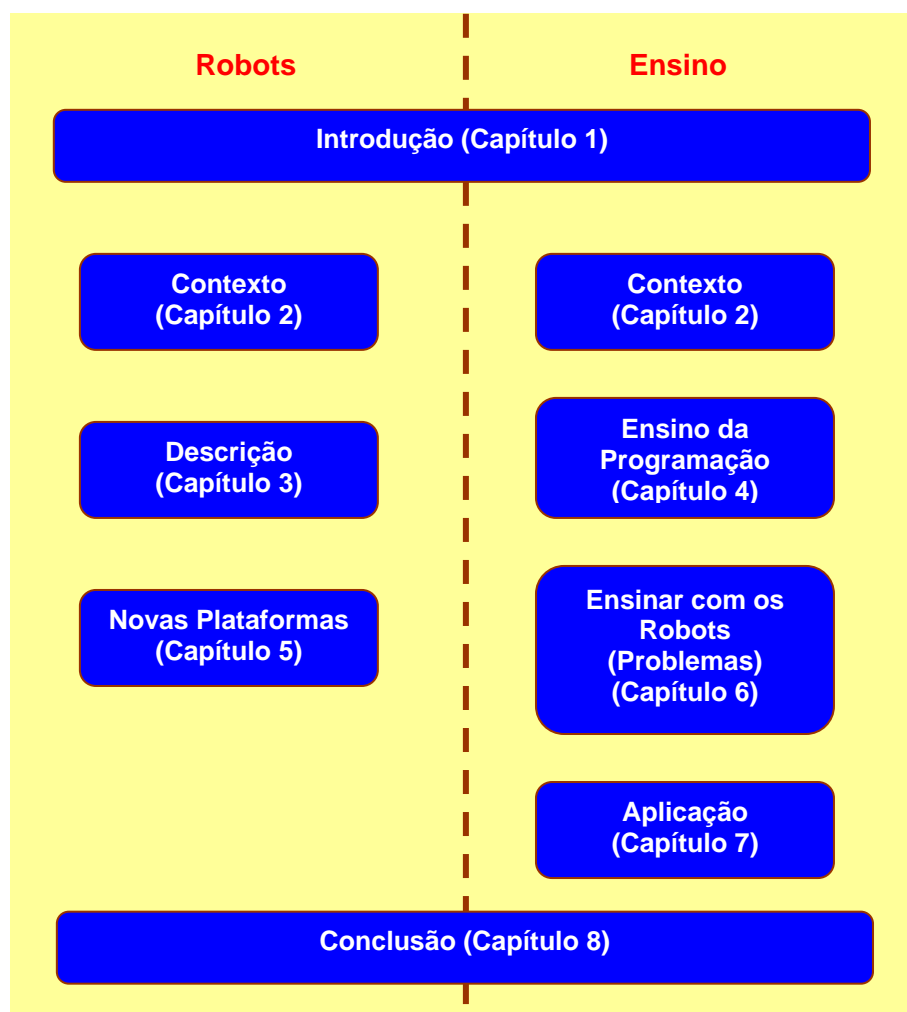


Figura 1 – Organização da Dissertação

O capítulo 1 “Introdução”: Contextualiza o trabalho de investigação, apresenta a motivação, objectivos e contributos. Por fim, sintetiza a organização da dissertação.

O capítulo 2 “Estado da Arte”: Apresenta o estado da arte da utilização dos robots no ensino da informática em geral, bem como as soluções existentes para a programação dos robots. Este capítulo pretende ainda enunciar algumas limitações e desafios na utilização dos robots no ensino.

O capítulo 3 “Descrição Teórica dos Robots”: Apresenta um estudo detalhado sobre o Kit da Lego® Mindstorms™, descrevendo os seus componentes, o firmware e linguagens de programação suportados por este.

O capítulo 4 “O Ensino da Informática em Portugal”: Apresenta um enquadramento do ensino da informática em Portugal, descreve ainda uma análise curricular dos programas das disciplinas de informática do ensino secundário e por fim apresentamos as razões da escolha da disciplina onde os robots serão usados.

O capítulo 5 “Novas Plataformas de Programação”: Apresenta as plataformas de programação criadas para programar os robots. Temos a plataforma RCX + Prolog, uma plataforma para programar os robots com a linguagem Prolog nas aulas de Inteligência Artificial, do Curso Engenharia Informática e a plataforma RCX+Pascal, uma plataforma para ser usada nas aulas de Bases de Programação do 10.º ano do ensino secundário.

O capítulo 6 “Problemas”: Apresenta uma série de problemas para serem usados nas aulas de Bases de Programação de 10.º ano, para serem resolvidos usando a linguagem pascal e os robots, através da plataforma RCX + Pascal apresentado no capítulo anterior.

O capítulo 7 “O Robot em Acção”: Descreve a aplicação no terreno de algumas das contribuições desta dissertação. Destaca-se neste capítulo a acção de formação realizada a professores de Informática e finalistas da Licenciatura em Ensino de Informática e a aplicação da Plataforma RCX + PROLOG, para a resolução do projecto de avaliação da disciplina de Introdução à Inteligência Artificial.

O capítulo 8 “Conclusões e Trabalhos Futuros”: Apresenta as conclusões desta dissertação bem como algumas propostas de trabalho futuro a desenvolver.



---

## 2. Estado da Arte

---

*“Study the past if you would define the future.”*

**Confucius**

Desde há alguns anos que vêm sendo desenvolvidas em várias partes do mundo, experiências com a utilização de Robots como uma ferramenta educativa, com especial incidência ao nível do ensino superior e em alguns casos no ensino básico ou secundário.

Este capítulo aborda na secção 2.1 o conceito da robótica educativa e na secção 2.2 apresenta alguns exemplos de interesse da utilização dos robots no ensino da informática. Na secção 2.3 apresentamos algumas limitações e desafios que surgiram nesta área de investigação e, finalmente, na secção 2.4 apresentamos a conclusão deste capítulo.

### 2.1. Robótica educativa

---

A introdução da Robótica no ensino pode parecer, à primeira vista, mais um conteúdo a ensinar ou a explicar aos alunos, numa perspectiva mais tradicional. Contudo, a robótica tem vindo a afirmar-se como uma ferramenta extremamente útil. O processo de construir robots ajuda os alunos na compreensão de conceitos sobre sistemas dinâmicos e complexos [OHM1999].

Perante isto, a pergunta que se coloca é a seguinte: qual a razão de integrar tecnologia na forma de robots no processo de ensino? Papert defende que, se este tipo de

tecnologia for empregue com eficácia, pode ter um impacto positivo no processo de ensino, que consequentemente melhorará o processo de aprendizagem. [P1980]

A abordagem construcionista defendida por Papert propõe a ideia de que os seres humanos aprendem melhor quando são envolvidos no planeamento e na construção de objectos ou artefactos que considerem significativos, partilhando-os com a comunidade envolvente. O processo de construção externa do objecto é, em paralelo, acompanhado da construção interior do conhecimento sobre o mesmo. [P1980]

A grande inovação em relação ao construtivismo passa, assim, pela valorização do papel das construções físicas como suporte das construções intelectuais. Os ambientes computacionais, e a Robótica em particular, constituem ferramentas poderosas para suportar estas novas formas de pensamento e de aprendizagem envolvendo os alunos no desenvolvimento de projectos significativos.

Chella definiu Robótica Educativa como um ambiente constituído pelo computador, componentes electrónicos, electromecânicos e programas, onde o aluno, através da integração destes elementos, constrói e programa dispositivos automatizados, com o objectivo de explorar conceitos das diversas áreas do conhecimento. [C2002]

De acordo com Zilli a robótica educativa, além de proporcionar aos alunos o contacto com a tecnologia actual, sugere o desenvolvimento do seguinte conjunto de competências: [Z2004]

- Raciocínio lógico;
- Habilidades manuais e estéticas;
- Relações interpessoais e intrapessoais;
- Utilização de conceitos aprendidos em diversas áreas do conhecimento para o desenvolvimento de projectos;
- Investigação e compreensão;
- Representação e comunicação;
- Trabalho com pesquisa;
- Resolução de problemas por meio de erros e acertos;
- Aplicação das teorias formuladas a actividades concretas;



- Utilização da criatividade em diferentes situações;
- Capacidade crítica.

A Robótica Educativa pode ser vista como uma ferramenta abrangente, que pode ser usada nos diversos níveis de ensino, e como forma de abordar diversos conteúdos sendo integrada no ensino numa perspectiva construtivista.

## **2.2. A utilização dos robots da Lego® Mindstorms™ no Ensino da Informática**

---

Nesta secção, pretende-se apresentar os principais projectos e iniciativas da utilização Robots da Lego® Mindstorms™ no ensino, mais concretamente no ensino da Informática, que foram sendo desenvolvidas ao longo dos últimos anos.

### **2.2.1. Competições**

Será incontestável afirmar que as competições ocupam um lugar de destaque. Este tipo de actividades exerce um grande fascínio, levando a níveis de participação e de entusiasmo normalmente bastante elevados. Neste contexto, salientam-se as seguintes iniciativas:

#### **First Lego League (FLL)**

Trata-se de uma competição (<http://www.firstlegoleague.org>) que envolve alunos dos níveis de ensino básico, com idades compreendidas entre os 9 e os 16 anos. Nasceu nos Estados Unidos de uma parceria entre a organização FIRST e a Lego®. A primeira competição data de 1992. Em cada ano a competição aborda temas distintos, tendo sido já tratados temas como os oceanos, a exploração de Marte ou a questão dos deficientes físicos. Em cada ano são lançados diversos desafios que as equipas (de 5 a 10 elementos) tentam resolver. No final de algum tempo (tipicamente cerca de 8 semanas) dedicados à construção dos robots usando os kits da Lego® Mindstorms™, as equipas disputam provas regionais e nacionais (inclusive Portugal), sendo as melhores seleccionadas para a final mundial.

## **RoboCup Júnior**

O projecto RoboCup (<http://www.robocupjunior.org>), com a parceria do Lego Lab na Dinamarca, a partir de 1999 organizou também competições destinadas a crianças entre os 9 e os 14 anos, num projecto que foi designado por RoboCup Jr. Nesta competição cada equipa tem dois robots autónomos que disputam um jogo de futebol contra outra equipa num campo que não pode ultrapassar os 3 metros.

Tipicamente, a plataforma Lego® Mindstorms™ serve de base aos robots mas estes podem ser adaptados com diferentes sensores e actuadores. Recentemente, existem diversas equipas a construir os seus robots de raiz.

O sucesso desta prova levou a que fossem aparecendo outras competições que foram sendo integradas no projecto. Uma delas, designada por salvamento, consiste num trajecto a ser percorrido pelos robots onde tem que ser seguida uma linha e no caminho alguns objectos (representando vítimas) têm que ser identificados, recolhidos e transportados para outro local (ou em alguns casos simplesmente identificados).

Em Portugal, têm-se realizado diversas provas do projecto RoboCup (<http://www.robocup2004.pt/>) tipicamente integradas nos festivais nacionais de Robótica que ocorrem desde 2001. O facto de se ter realizado a principal competição mundial em Portugal no ano de 2004, trouxe ainda mais animação a esta comunidade.

### **2.2.2. Projectos e Iniciativas**

Mas nem só em competições existe a utilização do robot no ensino da Informática. Existem outros projectos em todo o mundo que têm usado os robots da Lego para ensinar informática, alguns a nível curricular e outros, na grande maioria, em actividades extracurriculares. Apresentamos de seguida um conjunto de casos que achamos pertinentes e que ilustram os trabalhos realizados nesta área.

#### **Aprender Programação com RCX**

Wong usou os robots, RCX da Lego® nas aulas de programação da Universidade de Eastern kentucky. A ideia foi usá-los para facilitar o processo de aprendizagem de programação em três níveis diferentes (introdutório, intermédio e avançado). [W2001]

No nível introdutório, escolheram a disciplina de Programação em C++ onde usaram o robot para perceber o conceito de objectos e estruturas de controlo.

No nível intermédio, a disciplina escolhida foi Estruturas de Dados e foi pedido aos alunos para programar os robots de forma a evitar obstáculos e guardar a posição onde estes se encontravam.

No nível avançado, a disciplina escolhida foi Inteligência Artificial, onde os alunos tinham de implementar uma rede neuronal para resolver um determinado problema.

Em todos os níveis os alunos trabalharam em grupos e Wong concluiu que os robots são uma ferramenta poderosa para a aprendizagem da programação, pois permitiu-lhes, para além da aprendizagem dos conteúdos programáticos, aprender computação em tempo real, gestão de recursos e desenvolvimento de soluções modulares.

### **Aprender IA com Lego® Mindstorms™**

Parsons e Sklar usaram os robots da Lego® Mindstorms™ nas aulas de Inteligência Artificial e, pelas suas experiências, defendem que os robots são um benefício no processo de aprendizagem. [PS2004]

Utilizaram os robots na elaboração dos projectos, onde tinham grupos de 3 alunos ou mais, conforme o número de robots disponíveis. Os alunos tinham de montar o seu robot e resolver duas tarefas usando os sensores e o NQC como linguagem de programação.

Na primeira tarefa, o robot tinha de completar o mais rápido possível um trajecto marcado por uma linha preta. Ao longo do trajecto, o robot teria de remover obstáculos e identificar zonas coloridas emitindo um “beep”. Para terminar a tarefa, o robot tinha de identificar todas as áreas coloridas existentes no final do trajecto.

Na segunda tarefa, o ambiente não era fixo, como na anterior, e os alunos tinham de programar o seu robot para jogar futebol de um para um. Os robots eram colocados sobre um tabuleiro cinzento e a bola usada emitia luz infravermelha de forma a ser melhor identificada pelo sensor de luz.

Feito um inquérito aos alunos sobre qual das componentes tinham gostado e aprendido, o projecto foi, constantemente, o escolhido.

### **Aprender Java através do Lego® Mindstorms™**

David Barnes usou os robots da Lego® Mindstorms™ para ensinar Java numa disciplina de introdução à programação. [B2002]

Usou a API de Java existente para programar os robots e substituiu o firmware do robot pelo firmware alternativo LejOS (na secção 3.5 do capítulo seguinte pode ser consultada uma descrição deste firmware).

A utilização dos robots permitiu aos alunos identificar melhor os erros e conduzir a um raciocínio de programação mais eficaz.

Barnes conclui que os robots da Lego© Mindstorms™ são excelentes modelos físicos para simular os programas pelo facto de não ser necessário, tanto pelos alunos como pelos professores, grandes conhecimentos de hardware e por permitir interacção real com o meio ambiente, permitindo assim um feedback visível.

### **Aprender Linguagem C usando Robots**

Kim e Jeon criaram um curso de programação em C usando Lego® Mindstorms™ para principiantes em programação. [HW2005]

Usaram o IDE Bricx Command Center (pode consultar uma descrição deste IDE na secção seguinte deste capítulo) e o firmware alternativo BrickOS (descrito na secção 3.5 do capítulo seguinte).

Concluíram que os robots, juntamente com o IDE, são uma mais valia para os alunos, principalmente os que nunca programaram antes, pois passam de um ambiente de duas dimensões para um de três dimensões, tornando assim a junção dos robots com a linguagem C num excelente meio para ensinar a programar.

### **Projecto DROIDE UMa**

A utilização de robots como elementos mediadores entre o aluno e a Informática é um tema pouco estudado em Portugal, pelo menos no contexto de sala de aula. Assim, com o intuito de desenvolver investigação com a finalidade de compreender de que forma o uso da robótica contribui para que os alunos possam desenvolver a aprendizagem de tópicos e conceitos matemáticos e informáticos e, se possível a articulação entre as duas áreas de conhecimento, foi idealizado, no Departamento de Matemática e Engenharias da Universidade da Madeira, o projecto DROIDE: “Os Robots como elementos mediadores entre os alunos e a Matemática/Informática” iniciado no ano lectivo de 2005/2006 com a duração de três anos. [D2005]

O projecto tem como objectivos:

- Criar problemas na área da Matemática/Informática a serem resolvidos através dos robots;
- Criar robots para abordar problemas específicos na área da Matemática/Informática;
- Implementar a resolução de problemas utilizando robótica em três tipos de sala de aula:
  - de matemática no ensino básico e secundário;
  - de informática no ensino secundário;
  - de Inteligência Artificial (Licenciatura em Engenharia Informática), Didáctica da Informática (Licenciatura em Ensino da Informática) e Didáctica da Matemática (Mestrado em Ensino da Matemática), no ensino superior.
- Analisar a actividade dos alunos aquando da resolução dos problemas utilizando os robots nos diferentes tipos de aula referidos no II.

Em suma, este projecto pretende descrever, analisar e compreender como é que os alunos aprendem Matemática/Informática tendo os robots como elementos mediadores.

### **Projecto DROIDE VIRTUAL**

O projecto DROIDE VIRTUAL é um subprojecto do projecto DROIDE, apresentado anteriormente, iniciado em Outubro de 2006. [AFF2007]

O projecto inclui parceiros das Regiões da Madeira e Açores e de Portugal Continental, tendo como principal objectivo descrever, analisar e compreender o processo de construção do conhecimento acerca dos conceitos de programação e o desenvolvimento de competências nesta área, quando os alunos colaboram virtualmente na realização de projectos de robótica.

Os grupos envolvidos colaboram virtualmente no desenvolvimento de projectos de robótica, com o intuito de verificar a validade do uso de robots no ensino de programação num contexto de educação à distância, bem como analisar o papel das competições robóticas na motivação e no desenvolvimento da capacidade de trabalho em equipa.

## 2.3. Limitações e Desafios

---

A integração dos Robots no ensino da informática levanta algumas questões que podem constituir entraves à sua utilização.

Algumas dessas questões dizem respeito ao kit Lego® Mindstorms™, pois é necessário formar os professores no sentido de agilizar a utilização e manuseamento deste kit. É fundamental que o professor tenha conhecimento do seu funcionamento, pois são materiais com particularidades próprias.

Outra questão pertinente é a preparação dos professores para as metodologias a adoptar, uma vez que o trabalho a desenvolver está fora da prática que normalmente é desenvolvido na sala de aula. As metodologias a adoptar são claramente diferentes das práticas mais comuns.

Por outro lado, há a necessidade de esclarecer as pessoas, pois existe ainda uma ideia distorcida do que é a robótica. Ao falarmos de robótica quase invariavelmente se associa a ideia de ficção científica ou mesmo algo que se resume ao plano lúdico<sup>1</sup>.

## 2.4. Conclusão

---

Após a análise efectuada, chegamos às conclusões que a seguir sintetizamos.

A Robótica Educativa pode ser vista como uma ferramenta abrangente, que pode ser usada nos diversos níveis de ensino e como forma de abordar diversos conteúdos, sendo integrada no ensino numa perspectiva construtivista.

Apresentamos as duas vertentes onde a Robótica Educativa, usando o kit Lego® Mindstorms™ tem se destacado. Essas vertentes são as competições e os projectos e iniciativas nas salas de aulas.

As competições ocupam um lugar de destaque devido ao seu alto nível de participação, nas quais salientamos a First Lego League (FLL) e a RoboCup Júnior.

---

<sup>1</sup> A título de exemplo pessoal, muitas foram as situações em que, falando do trabalho que estava a ser desenvolvido, fomos confrontados com questões como: “Têm brincado muito com o robot?” ou “O robot consegue ajudar-vos lá em casa?”.

Quanto aos projectos e iniciativas apresentamos: (i) Casos de sucesso da utilização dos robots da Lego® Mindstorms™, nomeadamente: Aprender Programação com RCX [W2001], Aprender IA com Lego® Mindstorms™ [PS2004], Aprender Java através do Lego® Mindstorms™ [B2002] e Aprender Linguagem C usando Robots [HW2005]; (ii) Projectos inovadores em Portugal que têm como objectivo utilizar os robots como elementos mediadores entre o aluno e a informática, nomeadamente: Projecto DROIDE UMa [D2005] e o Projecto DROIDE VIRTUAL [AFF2007].

Para finalizar, citamos algumas reflexões sobre as limitações e desafios que podem surgir como consequência da integração dos Robots no ensino da informática.

No capítulo seguinte apresentamos uma descrição teórica do kit Lego® Mindstorms™.





---

## 3. Descrição Teórica dos Robots

---

*“One, a robot may not injure a human being, or through inaction, allow a human being to come to harm;*

*Two, a robot must obey the orders given it by human beings except where such orders would conflict with the First Law;*

*Three, a robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.”*

**Isaac Asimov, Laws of Robotics from I. Robot (1950)**

Este capítulo descreve o kit Lego® Mindstorms™ e as suas principais características. Na secção 3.1 apresentamos o kit Lego® Mindstorms™ e os seus componentes, na secção 3.2 os firmware's utilizados e suportados pelo RCX. Na secção 3.3 apresentamos uma descrição das ferramentas existentes no mercado para programar os robots da Lego® Mindstorms™ e ainda na secção 3.4 as linguagens de programação, e alguns exemplos. Por fim, na secção 3.5 faremos uma breve conclusão sobre o capítulo.

## 3.1. O Kit Lego® Mindstorms™

---

O kit Lego® Mindstorms™ é o produto central da linha de robótica da Lego® e é constituído por 718 peças, em que a maioria são peças técnicas como: eixos, rodas, rodas dentadas, conectores, elásticos e cabos. Contudo, existe um conjunto de peças específicas responsáveis pelo desenvolvimento dos robots. [FGH+2002] São elas:

- Um RCX (Robotics Command Explorer);
- Dois motores;
- Dois sensores de toque;
- Um sensor de luz;
- Uma torre de transmissão de infravermelhos – porta USB;
- Um manual;
- Um CD-ROM com o software Robotics Invention System 2.0.

### 3.1.1. O RCX

O RCX não é mais do que um pequeno computador, que interage com o exterior através de um pequeno LCD, três portas de entrada, três de saída, quatro botões e um módulo de comunicação infravermelhos, como mostra a figura seguinte. [FF2002]



Figura 2 - O RCX da Lego MindStorms

O RCX é mais semelhante a um computador industrial, criado para controlar máquinas, do que a um computador normal. É baseado na série de microprocessadores Hitachi H8, equipado com memória e temporizadores. A tabela seguinte resume as especificações do Hardware que o compõe. [FGH+2002]

Processador	8-bit Hitachi H8 Series
ROM	16 Kb
SRAM - Interna	512 bytes
SRAM - Externa	32 Kb
Saída	3 Portas para os motores de 9V 500 mA
Entradas	3 Portas para Sensores
LCD	1 LCD
Som	1 Unidade de Som
Temporizadores	4 Temporizadores de Sistema (8-bit)
Baterias	6 Baterias de 1.5V
Adaptador de Energia	9-12V, DC/AC (Apenas no RIS 1.0)
Comunicação	Porta de Infravermelhos (Transmissor + Receptor)

Tabela 1 - Especificações do Hardware do RCX

Para percebermos o funcionamento do RCX temos de imaginar uma estrutura em camadas, como representa a Figura. Na camada inferior temos o processador da série Hitachi H8, que executa as instruções da máquina e gere as entradas e saídas dos componentes adicionais.

Numa camada acima temos a ROM, que contém uma série de instruções, responsável por todas as funcionalidades básicas, como por exemplo: controlo de portas, funções do LCD, comunicações, entre outros. Podemos comparar esta camada à BIOS de um computador pessoal, pois é a responsável pelo arranque e pela comunicação entre os periféricos ou componentes. Sem esta camada não seria possível a interação do utilizador com o RCX.

Numa camada superior temos o firmware, que é uma espécie de sistema operativo para o RCX, que providencia o funcionamento de todo o sistema. Neste caso particular, esta camada é responsável pela interpretação dos programas em bytecode e por convertê-los

em instruções de máquina. Fica armazenado na RAM do RCX, pelo que deve ser feito o download, via torre de infravermelhos, a cada primeira instalação do sistema.

Finalmente, temos a camada dos Programas, onde ficam armazenados os programas criados no computador. Estes programas, antes de passarem do computador para o RCX, são convertidos, através do respectivo IDE, para um formato mais compacto e compatível com o processador do RCX, conhecido por programas em bytecode.

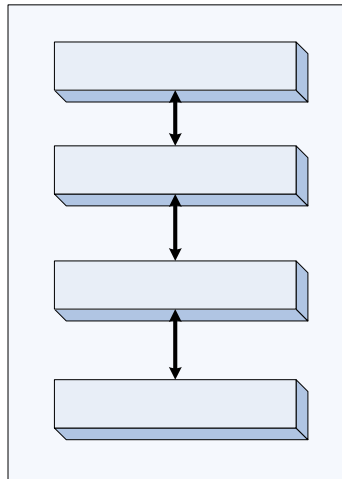


Figura 3 – A Estrutura Lógica do RCX

Quando o RCX é desligado a informação, o firmware e os programas, que são guardados na memória do RCX, não se perdem, uma vez que a RAM mantém-se ligada à energia fornecida pelas baterias. Mesmo estando desligado o RCX continua, embora que em quantidades pequenas, ligado à energia das baterias.

Programar o RCX é um processo simples que se resume aos seguintes passos:

- Escrever um programa num IDE do seu computador;
- Compilar o programa. As instruções do programa são transformadas em instruções de baixo nível – bytecode;
- Fazer o download do programa para o RCX através da torre de Infravermelhos;
- O firmware instalado no RCX interpreta o bytecode e converte-o para linguagem máquina;
- O processador executa as instruções.

### 3.1.2. Os Motores

Podemos ligar até 3 motores, usando um cabo específico com comportamento independente, ao nosso RXC nas portas A, B e C, respectivamente.

Os motores, ilustrados na figura seguinte, são usados pelos robots para se deslocarem, para mover “braços” ou noutra estrutura que se movimente. Os motores serão então a fonte principal de energia do robot, uma vez que a sua função é transformar energia eléctrica em energia mecânica.



Figura 4 – Motor

Os motores foram concebidos para girar a uma voltagem de 9V. Contudo, são toleráveis à diminuição desta voltagem e, quando tal acontece, simplesmente giram mais devagar.

### 3.1.3. Os Sensores

Existem basicamente quatro tipos de sensores que podem ser ligados a uma das três portas: sensor de luz, sensor de toque, sensor de rotação e sensor de temperatura.

Os sensores podem ser classificados em dois tipos: os activos e os passivos. Os passivos são os que não necessitam de energia para funcionar, como o sensor de toque e o de temperatura. Por sua vez o sensor de luz e o de rotação pertencem ao tipo activo.

Os sensores são ligados ao RCX utilizando o mesmo tipo de cabo usado para ligar os motores.

É importante referir que os sensores de rotação e temperatura não fazem parte do Kit original, mas podem ser facilmente adquiridos.

## Sensor de Luz

O sensor de luz mede a intensidade de luz num determinado ponto. O sensor de luz emite também uma luz, que possibilita apontar o sensor numa determinada direcção e medir as diferenças da intensidade de luz num determinado objecto.

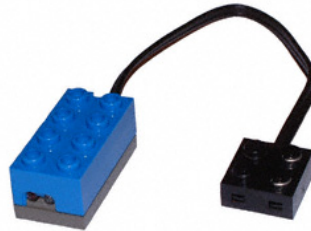


Figura 5 – Sensor de Luz

As medidas feitas pelo sensor são guardadas em percentagem (0 a 100%, sendo mais comuns os valores entre 40 e 60). O RCX utiliza essas medidas para determinar qual a luminosidade do ambiente em que o robot se encontra.

Uma utilização muito frequente para o sensor de luz é a distinção das cores da superfície baseado na capacidade de reflexão de luz. Como a luz reflectida depende da distância e da cor da superfície, mantendo a distância podemos distinguir as várias cores. Por exemplo, se tivermos uma linha preta sobre um fundo branco, quando o sensor estiver sobre a linha preta receberá menos luz reflectida do que no fundo branco. Desta forma, o RCX pode determinar se o sensor está ou não sobre a linha preta e daí efectuar alguma operação.

## Sensor de Toque

Podemos comparar o sensor de toque a um interruptor, que se encontra aberto e é fechado quando pressionado.



Figura 6 - Sensor de Toque

Na prática, o sensor de toque é usado apenas para distinguir estados, pelo que o valor da tensão exercida pelo pressionar não interessa, a menos que seja indicado. O RCX atribui-lhe dois estados: 0 para quando está aberto e 1 para quando está fechado ou pressionado.

É possível ligar mais do que um sensor de toque na mesma entrada para o caso de pretendermos que o robot faça a mesma acção para duas situações diferentes. Por exemplo, quando bate com a frente ou a traseira, sem ter de ocupar necessariamente duas portas. Os sensores ligados à mesma porta são capazes de detectar a acção em qualquer um deles, embora não seja possível diferenciar qual deles foi activado.

### **Sensor de Rotação**

O sensor de rotação consegue detectar rotações de 1/16, ou seja, 22,5°, bem como, distinguir o sentido da rotação.



Figura 7 - Sensor de Rotação

Contudo, este sensor tem o inconveniente de necessitar de calibração. Outro inconveniente diz respeito ao facto de haver atrito no próprio sensor, o que pode perturbar, de alguma forma, a medida efectuada.

### **Sensor de Temperatura**

Este sensor, dependendo da forma como é configurado, retorna valores de temperatura em graus Celsius ou Fahrenheit. O sensor é capaz de medir temperaturas desde -20 °C a 70°C ou de -4°F a 158°F.



Figura 8 - Sensor de Temperatura

Contudo, este sensor mostra-se com um desempenho fraco, uma vez que é muito lento a mudar de um valor para outro, pelo que pode não ser o melhor se se pretende medir mudanças repentinas de temperatura.

### 3.1.4. A Torre de Transmissão

Para enviar ao RCX o firmware ou programas, é necessário um computador ao qual se liga, via porta USB, uma torre de infravermelhos (ilustrado na figura seguinte). É esta torre que envia para o receptor de infravermelhos existente no RCX toda a informação.

Toda a comunicação entre o computador e o RCX é efectuada via infravermelhos, utilizando o emissor/receptor presente no RCX e na torre.



Figura 9 – Torre de Infravermelhos

As comunicações entre a torre de infravermelhos e o RCX estão sujeitas às limitações da tecnologia dos infravermelhos, pelo que a presença de um obstáculo opaco entre o emissor e o receptor pode comprometer a comunicação.

De qualquer forma, é possível configurar a torre no sentido de emitir para um alcance superior, o que requer maior potência na emissão do sinal e, conseqüentemente, maior consumo de energia fornecida pela porta USB do computador ao qual a torre está ligada.

## 3.2. O Firmware

---

O firmware é o sistema operativo (SO) para o RCX que providência o funcionamento de todo o sistema. É responsável pela interpretação dos programas em bytecode e convertê-los em instruções máquina.

Fica armazenado na RAM do RCX, pelo que deve ser feito o download, via torre de infravermelhos, a cada primeira instalação do sistema.

Segue-se uma breve descrição dos principais firmware existentes para o RCX.



### **3.2.1. LEGO**

O firmware da Lego®, ou firmware padrão, é um sistema operativo simples, constituído basicamente por uma máquina virtual, que apresenta uma grande estabilidade. [DVL2004]

Uma das vantagens da máquina virtual é a abstracção da arquitectura, pois se a arquitectura for alterada o firmware pode continuar a ser compatível com os programas escritos antes das alterações.

Tem como tarefa básica a interpretação do bytecode e transformá-los numa série de instruções e dados apropriados aos microprocessadores Hitachi H8 do RCX.

Entre os IDE's que geram bytecode para o firmware da LEGO destacam-se o Robolab e o RIS apresentados na secção seguinte.

### **3.2.2. BrickOS**

O BrickOS foi desenvolvido originalmente por Markus L. Noga, cuja primeira versão tinha o nome LegOS. [BOS] O nome foi mudado mais tarde, a pedido da empresa Lego®.

O BrickOS é um sistema operativo alternativo para o RCX. É um sistema operativo multi-tarefa, que permite a gestão dinâmica da memória, semáforos, secções críticas e um acesso nativo ao LCD, aos botões, à comunicação por infravermelho, aos motores e aos sensores.

Suporta as linguagens de programação C, C++ e Pascal.

É um produto open source (disponível no endereço <http://brickos.sourceforge.net/>), suportado na maioria dos sistemas operativos de Linux e Windows, através do IDE BricxCC, apresentado na secção seguinte.

### **3.2.3. LejOS**

O LejOS é também firmware [LOS] alternativo para o RCX. É uma Java Virtual Machine (JVM) extremamente compacta, ocupando cerca de 10 Kb no RCX, e que actua como um sistema operativo.

O LejOS providencia uma API (Application Programming Interface), ou seja, uma biblioteca que permite aceder directamente às entradas e saídas do RCX. Esta API oferece ainda métodos para controlar o RCX com uma implementação fácil. Permite controlar os sensores de luz e de toque, os motores e a torre de infravermelhos.

Este firmware surgiu quando Jose Solorzano começou há uns anos o projecto TinyVM. O seu objectivo era construir JVM o mais compacta possível mas que suportasse as mais complexas evoluções de um sistema Java.

É um produto open source (disponível no endereço <http://lejos.sourceforge.net/>), suportado nos sistemas operativos Windows, através do IDE BricxCC.

### 3.3. Ferramentas de Programação Existentes

---

As ferramentas ou IDE de programação são essenciais no processo de programação do robot. Estas são o elo de ligação entre o aluno, a programação e o robot. A seguir apresentamos uma breve referência às principais ferramentas de programação do RCX, nomeadamente, o Robotics Invention System™, o ROBOLAB e o Bricx Command Center.

#### 3.3.1. Robotics Invention System

O Robotics Invention System™ 2.0, ou simplesmente denominado por RIS, é uma aplicação que permite programar o RCX arrastando blocos de código para a área do programa.[RIS]

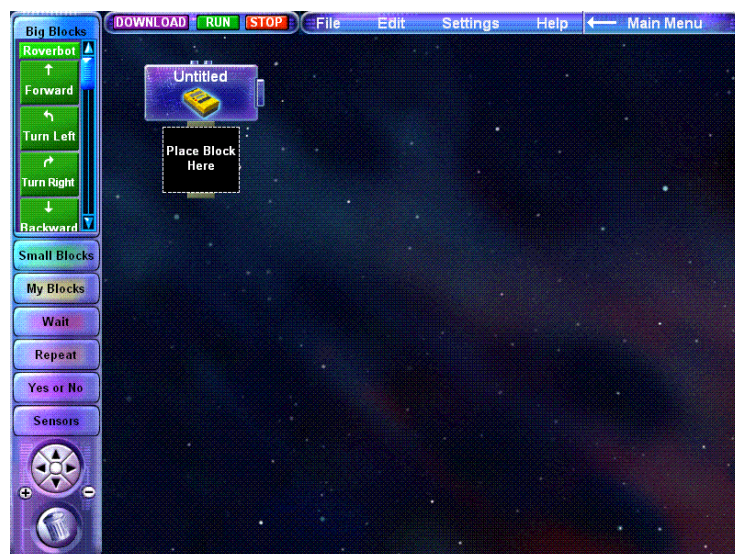


Figura 10 - Ambiente de programação do RIS

O RIS oferece-nos um tipo de programação muito intuitiva, orientada a eventos.

Ao nosso dispor temos os “Blocos Grandes” que têm, completamente, definidos os movimentos do robot e algumas funcionalidades associadas aos sensores. Existem também os “Blocos Pequenos”, que permitem controlar potência dos motores, variáveis, saída do LCD e sons.

Temos também à nossa disposição blocos para instruções “Repetir” ou “Esperar Por”, “Sim-Então-Senão”, entre outras.

A grande vantagem do RIS deve-se ao facto da programação do robot ser feita por arrasto de blocos, de uma forma visual, evitando erros de sintaxe. Esta particularidade é uma ferramenta muito indicada para pessoas que não têm grandes conhecimentos de programação.

Outra das vantagens é o facto de ser possível, na estrutura de ficheiros da aplicação, aceder à biblioteca da linguagem e traduzir quase na totalidade a aplicação. Esta é, sem dúvida, uma mais valia, pois se estiver em português, pode deixar de ser um entrave para quando estamos a lidar com alunos que não possuam conhecimentos de inglês.

### **3.3.2. ROBO LAB™**

O ROBO LAB™ é uma aplicação que disponibiliza um ambiente de programação e ferramentas para programar, aprender e explorar o RCX. [MSE+2002]

É uma aplicação caracterizada, tal como o RIS, por ter uma interface gráfica intuitiva, em que os comandos para programar o robot estão representados de forma gráfica, bastando apenas ligá-los de uma forma lógica conforme a solução pretendida.

A figura seguinte mostra o ambiente de programação do ROBO LAB™.

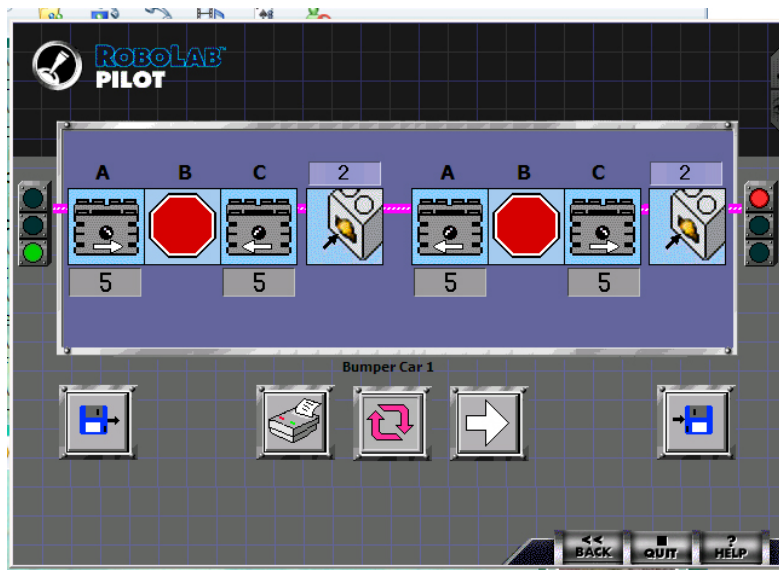


Figura 11 - Ambiente de programação do ROBOLAB™

O ROBOLAB™ é construído sobre um software robusto, cujo nome é LabVIEW™. [LVIEW] O LabVIEW™ é um ambiente de programação poderoso usado por engenheiros e cientistas em universidades e indústrias.

É o líder entre o software de ferramenta de desenvolvimento para medidas e controlo. Foi criado pela National Instruments (Texas, US) em 1997.

### 3.3.3. Bricx Commander Center

Mark Overmars criou o Bricx Command Center (BricxCC) [BCC] que, pelo menos na versão 3.3, permite programar o RCX e outros kits semelhantes (Cybermaster, Scout, Spybot e Microscout).

O BricxCC é uma aplicação que corre sobre Windows e é classificado como um IDE de programação.

Enquanto que no RIS a programação é feita através da combinação de blocos, no BricxCC é feita através de linhas de código escritas, o que a direcciona para um público-alvo que possui alguns conhecimentos de programação.

A figura seguinte mostra o ambiente de programação do BricxCC.

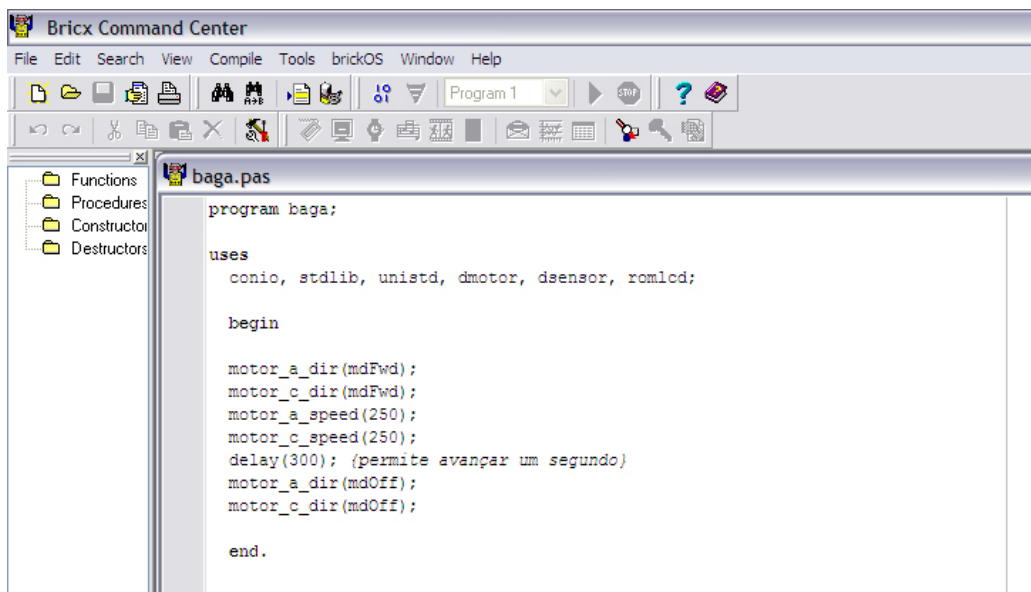


Figura 12 – Ambiente de programação do BricxCC

Esta ferramenta suporta a programação do RCX em linguagem C, C++, Pascal e Java, usando firmware alternativos, nomeadamente o brickOS e leJOS. No capítulo 3, secção 3.5, serão abordados os firmware do RCX.

A vantagem da utilização do BricxCC deve-se ao facto do utilizador ter à sua disposição a possibilidade de poder compilar e depurar o programa, prática essencial na aprendizagem da programação. O utilizador tem também à sua disposição várias opções, que vão desde a configuração da porta onde está ligada a torre de infravermelhos até comandar o RCX remotamente.

## 3.4. Linguagens de Programação

---

Os Robots Lego® MindStorms™ têm uma grande comunidade de fãs e alguns deles têm vindo a desenvolver, ao longo dos tempos, diferentes formas de programar o RCX. Criam “clones” das linguagens de programação existentes, com as quais se sentem mais à vontade, ou até criam novas linguagens.

Neste sentido, podemos classificar as linguagens de programação em dois grupos: linguagens nativas e linguagens não nativas. As linguagens nativas são todas as linguagens que correm sobre o firmware LEGO, apresentado na secção anterior. As linguagens não nativas são todas as linguagens que necessitam de firmware alternativo, como o caso do BrickOS e LeJOS, para poderem ser executadas no robot.

No primeiro grupo encaixa apenas a linguagem NQC, enquanto que no outro grupo o leque de linguagens é superior. A seguir apresentamos as linguagens de programação, existentes para os robots.

### 3.4.1. Linguagem Nativa – NQC

NQC, ou Not Quite C, é uma linguagem simples para programar os robots da Lego MindStorms. O compilador do NQC traduz o código fonte em LEGO bytecodes, que pode ser então executado no RCX sobre o firmware LEGO, apresentado na secção 3.2.1 deste capítulo. Todas as estruturas da linguagem e o pré-processador são muito semelhante à linguagem C. [B2003]

Um dos pontos fortes do NQC é o facto de utilizar o mesmo firmware que os produtos da Lego, o que nos possibilita, e apenas dando um exemplo, ter ao mesmo tempo no RCX programas feitos com o NQC e com o IDE RIS.

Como já foi referido, a sintaxe do NQC é muito parecida com a sintaxe da linguagem C. Contudo, para uma melhor aprendizagem da linguagem, e principalmente das funções específicas que dizem respeito ao robot, sugerimos a consulta do guia de programação do NQC de Dave Baum disponível na página oficial (<http://bricxcc.sourceforge.net/nqc/>).

A tabela seguinte mostra um pequeno exemplo de um programa escrito na linguagem NQC. O programa faz com que o RCX avance durante quatro segundos e recue durante dois segundos.

```
task main()  
{  
  OnFwd(OUT_A);  
  OnFwd(OUT_C);  
  Wait(400);  
  OnRev(OUT_A+OUT_C);  
  Wait(200);  
  Off(OUT_A+OUT_C);  
}
```

Tabela 2 - Exemplo de um programa escrito na linguagem NQC.

### 3.4.2. Linguagens não Nativas

#### Java

A “API RCX Java” é uma API de alto nível que possibilita a comunicação com o RCX através da torre de infravermelhos [ARJ]. Esta biblioteca esconde os pormenores de baixo nível do protocolo de comunicação, proporcionando assim uma interface simples de utilizar na programação do robot.

A Tabela seguinte mostra um pequeno programa escrito em Java, que utiliza algumas das funções disponibilizadas pela “API RCX Java”.

```
import rcx.*;

public class Test {
    public static void main(String[] arg) {
        new Test(arg[0]);
    }
    public Test(String portname) {
        RCXPort port = new RCXPort(portname);
        Motor.A.forward();
        Motor.A.stop();
        port.beep();
        System.out.println(
            "Battery Level = "+port.getBatteryPower()+" volts.");
    }
}
```

Tabela 3 - Exemplo de um programa escrito na linguagem Java

Este programa activa o Motor A por um tempo predefinido de 1 segundo. Depois para o motor e o RCX emite um “beep”. Para finalizar mostra o nível da bateria.

A API RCX Java oferece também uma Applet que controla o RCX, desde que esteja ao alcance da torre de infravermelhos ligada a um computador remoto.

#### Pascal

A API de Pascal tem um conjunto de funções definidas que permite programar o robot usando a linguagem Pascal. O código é escrito e compilado no IDE BricxCC e para poder ser executado no robot tem de ser instalado no RCX o firmware BrickOS. No anexo A está disponível um manual da API de Pascal.

A tabela seguinte mostra um pequeno programa escrito em Pascal. O programa permite avançar o Motor A e C do robot em simultâneo durante dois segundos.

```

program avançal;

uses
  unistd, dmotor;
begin
  motor_a_dir(mdFwd);
  motor_c_dir(mdFwd);
  motor_a_speed(255);
  motor_c_speed(255);
  sleep(2);
  motor_a_dir(mdOff);
  motor_c_dir(mdOff);
end.

```

Tabela 4 - Exemplo de um programa escrito na linguagem Pascal

### C/C++

A API de linguagem C/C++ tem um conjunto de funções definidas que permite programar o robot usando a linguagem C ou C++. Tal como na API de Pascal, o código é escrito e compilado no IDE BricxCC e para poder ser executado no robot também tem de ser instalado no RCX o firmware BrickOS.

## 3.5. Conclusão

---

Depois da análise e estudo do kit Lego® Mindstorms™, apresentado ao longo deste capítulo, concluímos que é uma ferramenta muito flexível, tanto a nível de construção dos robots e utilização dos seus componentes, como a nível de software para a programação do RCX.

Para além das inúmeras vantagens associadas a este kit, queremos destacar a facilidade de construção do robot (podemos dizer que depende da imaginação de cada um) e a sua flexibilidade na programação.

Apresentamos os firmware's suportados pelo RCX, bem como as linguagens de programação. O facto de ser possível programar o robot com diversas linguagens, abre-nos as portas para podermos usar este kit em disciplinas específicas de programação.

Apresentamos ainda as ferramentas de programação existentes no mercado para programar os robots, onde se destacam o RIS e o ROBOLAB pela sua filosofia de programação, bastando arrastar blocos de código para a área do programa, e o BricxCC



por permitir a programação através de linhas de código escritas e suportar as linguagens de programação Pascal, C, C++ e Java.

No capítulo seguinte fazemos uma análise ao Ensino da Informática em Portugal e analisamos quais as disciplinas onde podemos aplicar os robots aqui apresentados.



---

## 4. Ensino da Informática em Portugal

---

*“Education's purpose is to replace an empty mind with an open one.”*

**Malcolm Forbes**

Este capítulo apresenta um enquadramento do Ensino da Informática em Portugal no Ensino Secundário. Na secção 4.1 apresentamos a finalidade do Ensino da Informática, na secção 4.2 apresentamos uma análise curricular das disciplinas do Curso Tecnológico de Informática. Apresentamos ainda, na secção 4.3, as razões da escolha da disciplina onde os robots serão usados.

### 4.1. A Finalidade do Ensino de Informática

---

O sistema educativo português assume como objectivo estratégico a necessidade de assegurar a todos os jovens o acesso às tecnologias da informação e comunicação, como condição indispensável para a melhoria da qualidade e da eficácia da educação e formação, à luz das exigências da sociedade do conhecimento.

A disciplina de TIC vem dar resposta a esta necessidade, garantindo a todos os alunos o domínio de um conjunto de competências e conhecimentos básicos em TIC e promovendo a integração, a articulação e o desenvolvimento das aprendizagens nesta área de formação. [J2003]

Contudo, áreas de conhecimento como a Análise, Programação, Tecnologias Informáticas e Bases de Dados, são mais valias significativas que se usam em diferentes áreas de engenharia e não só.

Baseando-se neste pressuposto o Curso Tecnológico de Informática oferece uma série de disciplinas com a finalidade de:

- Acrescentar mais valias no domínio da aplicabilidade da informática ao contexto real do quotidiano;
- Conduzir o aluno à resolução de problemas de uma forma estruturada e sistematizada, analisando todos os elementos intervenientes;
- Proporcionar um conjunto de experiências e vivências que permita aos alunos acompanhar a permanente evolução tecnológica, quer profissionalmente, quer em termos pessoais e humanos.

## **4.2. Análise Curricular dos Programas**

---

Para percebermos em que disciplinas poderiam ser aplicados os robots na resolução de problemas, fizemos uma análise curricular aos programas homologados pelo Ministério da Educação, das várias disciplinas do Curso Tecnológico de Informática do Ensino Secundário. Os programas analisados foram os seguintes:

- Aplicações Informáticas A 10.º ano [JPD2004];
- Aplicações Informáticas A 11.º ano [PDJ2005];
- Bases de Programação 10.º, 11.º e 12.º anos [GSP+2002];
- Tecnologias Informáticas 10.º e 11.º anos [SCF2001];
- Técnicas de Gestão de Bases de Dados do 12º ano [PSD+2006].

### **4.2.1. Aplicações Informáticas A - 10.º ano**

A disciplina de Aplicações Informáticas A 10.º ano é uma disciplina de carácter eminentemente prático, onde se pretende que os alunos desenvolvam capacidades para

utilizar adequadamente, e manipular com rigor técnico, um conjunto vasto de aplicações informáticas.

Como se trata de um programa de aplicações, pressupõe-se que o aluno aplique sistematicamente as aprendizagens feitas, de acordo com uma metodologia de aprendizagem activa, coordenando a vertente conceptual e operacional dos conteúdos e integrando-as num contexto de posto de trabalho.

As principais finalidades a desenvolver ao longo da disciplina no 10º ano são:

- Promover a utilização das tecnologias de informação e comunicação numa perspectiva profissional integrada;
- Desenvolver capacidades para usar adequadamente, e manipular com rigor técnico, aplicações informáticas;
- Estimular hábitos de análise crítica à estrutura global de Aplicações Informáticas;
- Promover hábitos de auto aprendizagem nos ajustamentos e evolução permanente de software;
- Estimular a capacidade de procura nas soluções informáticas das ferramentas de sistema capazes de resolver problemas profissionais;
- Estimular os alunos a examinar criticamente a função e o poder das tecnologias de informação, numa vertente profissional e analítica das aplicações;
- Promover o interesse pela ciência, tecnologias e pela investigação científica e tecnológica, numa perspectiva de formação permanente;
- Promover o desenvolvimento de competências de trabalho em equipa e responsabilidade individual.

Esta disciplina pretende desenvolver nos alunos as seguintes competências:

- Demonstrar destreza no trabalho em ambientes gráficos;
- Identificar e resolver problemas de utilização de sistemas operativos;
- Manipular documentos de diferentes tipos e dimensões numa perspectiva de produção para utilizador;

- Manipular a organização da informação numa perspectiva integrada de descrição e cálculo;
- Assumir responsabilidades de trabalho individual e em equipa;
- Manifestar disponibilidade para aprofundar a sua formação no contexto da sociedade de reformulação.

### **4.2.2. Aplicações Informáticas A - 11.º ano**

O 11º ano da disciplina de Aplicações Informáticas A é o ano terminal desta disciplina no contexto curricular do curso.

Esta disciplina cobre duas componentes de aprofundamento de saberes específicos e duas de introdução, a áreas que sendo essenciais possuem níveis e especificidades diferentes e inovadoras.

Em relação às primeiras duas componentes o programa cobre a área de concepção de Web Sites e a área de Multimédia. Já no que diz respeito às componentes inovadoras a disciplina cobre a área de Análise de Sistemas e Inteligência Artificial (IA).

As principais finalidades a desenvolver ao longo da disciplina no 11º ano são:

- Identificar os processos fundamentais da informação digital;
- Aplicar as metodologias da análise estruturada de sistemas;
- Utilizar a modelação como ferramenta;
- Operacionalizar e implementar o conceito de IA;
- Utilizar conhecimentos relativos às lógicas estruturais e aos modos de interacção de aplicações multimédia na análise da sua utilidade, do seu interesse e da sua eficácia;
- Avaliar a eficácia e funcionalidade de software multimédia;
- Criar e publicar páginas na Web, usando editores e ferramentas de cariz profissional;
- Realizar projectos interdisciplinares utilizando os procedimentos da metodologia de trabalho de projecto;

- Criar e mostrar produtos construídos e/ou modificados por ferramentas multimédia.

### **4.2.3. Bases de Programação 10.º, 11.º e 12.º anos**

A disciplina Bases de Programação é uma disciplina trienal que pretende conduzir o aluno à resolução de problemas de uma forma estruturada e sistematizada, analisando todos os elementos intervenientes num determinado problema.

No 10º ano, o programa incide sobre a aprendizagem de algoritmia e programação estruturada em linguagem Pascal. Na aprendizagem da linguagem de programação Pascal deve-se executar programas de processamento variado, dando assim a possibilidade a que os alunos, nos anos posteriores, estejam aptos a resolver situações complexas.

No que se refere à disciplina de 11º ano é usada a linguagem Visual Basic, em que o aluno deverá adquirir não só as bases da programação por objectos, como a capacidade de elaborar pequenas aplicações informáticas utilizando uma ferramenta visual.

No 12º ano é usada uma linguagem de nível mais avançado, o Visual C++. Através desta linguagem é pretendido a consolidação e alargamento de conhecimentos na área das linguagens de programação. Haverá ainda espaço para o estudo genérico de, pelo menos, uma linguagem de programação que se considere adequado às exigências informáticas do momento (por ex. JAVA, PHP, ASP, PERL), preparando desta forma o aluno para o estudo de novas linguagens.

Ao longo dos três anos, a disciplina de Bases de Programação tem como finalidades:

- Desenvolver a capacidade de reflexão, gerando um saber criativo e evolutivo;
- Desenvolver a autoconfiança com base nos saberes e competências adquiridos;
- Fomentar o interesse pela investigação;
- Proporcionar um conjunto de experiências e vivências que permitam aos alunos acompanhar a permanente evolução tecnológica, quer profissionalmente, quer em termos pessoais e humanos;

- Promover o espírito crítico, a capacidade de cooperação, o respeito pelos outros e a responsabilidade, permitindo que os alunos se desenvolvam integralmente enquanto indivíduos e enquanto cidadãos;
- Desenvolver competências de trabalho em equipa;
- Promover a aquisição de competências e versatilidade no desenvolvimento de aplicações informáticas;
- Promover o respeito pela propriedade intelectual.

Esta disciplina pretende desenvolver nos alunos as seguintes competências:

- Identificar os diversos factores intervenientes num problema;
- Utilizar diferentes estratégias de tratamento dos dados;
- Fasear a resolução de um problema;
- Resolver problemas, encontrando soluções criativas e adequadas;
- Usar metodologias de análise algorítmica;
- Utilizar diferentes métodos algorítmicos;
- Utilizar estruturas de decisão;
- Utilizar estruturas de repetição;
- Criar algoritmos estruturados;
- Adaptar-se à evolução das linguagens.

#### **4.2.4. Tecnologias Informáticas 10.º, 11.º e 12.º anos**

A disciplina Tecnologias Informáticas do Curso Tecnológico de Informática é uma disciplina trienal que pretende proporcionar aos alunos conhecimentos fundamentais sobre a utilização de computadores e as suas diferentes arquitecturas. Pretende, também, despertar uma atenção permanente às mudanças contínuas que se sentem nas Tecnologias da Informação e Comunicação.

No 10.º ano pretende-se fornecer aos alunos os conhecimentos básicos na área das Tecnologias da Informação, equipamentos e realizar uma pequena abordagem aos Sistemas Operativos.



No 11.º introduzir alguma formalidade e rigor nos conteúdos abordados no ano anterior, com a Teoria dos Sistemas Operativos e das Redes de Computadores, fazendo ao mesmo tempo uma incursão em Sistemas Operativos de Rede baseados em DOS, NFS e NTFS, nos quais aplicarão a teoria desenvolvida.

No 12.º pretende-se que o aluno, por um lado, conheça a arquitectura Macintosh como alternativa aos PCs e, por outro lado, aprofundar os conhecimentos de redes abordando o sistema operativo UNIX e os seus “clones”. Pretende-se ainda neste ano que o aluno explore sistemas operativos de grande porte.

A disciplina de Tecnologias Informáticas tem como finalidades:

- Desenvolver a capacidade de reflexão, gerando um saber criativo e evolutivo;
- Desenvolver a autoconfiança com base nos saberes e competências adquiridas;
- Proporcionar um conjunto de experiências e vivências que permitam aos alunos acompanhar a permanente evolução, quer profissionalmente, quer em termos pessoais e humanos;
- Promover o espírito crítico, a capacidade de cooperação, o respeito pelos outros e a responsabilidade, permitindo que os alunos se desenvolvam integralmente enquanto indivíduos e enquanto cidadãos;
- Promover a aquisição de competências e versatilidade na utilização de sistemas operativos e equipamentos de redes.

Esta disciplina pretende desenvolver nos alunos as seguintes competências:

- Conhecer os conceitos básicos e terminologias inerentes às Tecnologias da Informação e Comunicação, bem como a evolução desses mesmos conceitos;
- Dominar metodologias de análise de sistemas informáticos;
- Instalar, configurar e operar correctamente diferentes sistemas operativos;
- Adaptar-se às novas tecnologias, que irão surgir no decorrer da sua vida activa;
- Cumprir normas de segurança de informação, assegurando a manutenção e segurança dos sistemas operativos;

- Decidir pela melhor das soluções na realização de um problema concreto;
- Usar vários suportes de informação para resolver dificuldades concretas na realização de qualquer tarefa.

#### **4.2.5. Técnicas de Gestão de Bases de Dados 12º Ano**

No 12º ano de escolaridade a disciplina Técnicas de Gestão de Bases de Dados constitui uma das especificações terminais da componente de formação tecnológica do Curso Tecnológico de Informática.

Neste contexto, surge sob a forma de uma pré-especialização, que deverá ter continuidade no mundo do trabalho.

Sendo o Curso Tecnológico de Informática fortemente orientado para a vida activa é natural que as suas componentes terminais ao nível do ensino secundário estejam dirigidas para o que possam ser as prováveis orientações de aplicabilidade no contexto de trabalho.

Nos termos dos pressupostos orientadores enquadrados pelo Decreto-Lei n.º 74/2004, de 26 de Março, deve ainda esta disciplina articular-se com as aprendizagens desenvolvidas na Área Tecnológica Integrada (ATI), nomeadamente com o Projecto Tecnológico (PT) e o Estágio, e contribuir de forma coerente e adequada para a Prova de Aptidão Tecnológica (PAT).

A disciplina de Técnicas de Gestão de Bases de Dados tem como finalidades:

- Promover o incremento das capacidades de produção colaborativa, entre as quais se salientam a co-criatividade e a co-responsabilidade, numa perspectiva de abertura à mudança, de compreensão dos fenómenos mediáticos e de percepção do papel dos conteúdos nas TIC;
- Desenvolver capacidades necessárias à manipulação de aplicações informáticas, em articulação com as aprendizagens de carácter específico e de outras áreas de formação inseridas no currículo;
- Criar hábitos e atitudes conducentes a uma disponibilidade para uma aprendizagem ao longo da vida, como condição essencial exigida para a adaptação a um crescimento acelerado de novas formas de comunicar, que

continuamente criam novos afloramentos do saber associados ao contexto da sociedade do conhecimento;

- Desenvolver capacidades para usar adequadamente e manipular com rigor técnico programas gestores de bases de dados;
- Desenvolver capacidades para gerir e manter informação de bases de dados;
- Sensibilizar os alunos para a importância da segurança e da privacidade de dados;
- Promover as práticas inerentes à segurança, higiene e saúde no trabalho que estejam relacionadas com os condicionalismos das profissões da área da informática, nomeadamente a ergonomia e a saúde ocular;
- Contribuir para a elaboração da Prova de Aptidão Tecnológica (PAT).

### **4.3. Onde aplicar os Robots**

---

Como resultado da análise efectuada consideramos que a disciplina Bases de Programação 10.º, 11.º e 12.º anos [GSP+2002] vai ao encontro dos objectivos estabelecidos para a aplicação de resolução de problemas usando os robots na aprendizagem de conceitos de programação.

A disciplina de Bases de Programação tem como objectivos gerais e essenciais fazer com que o aluno desenvolva raciocínio lógico em termos de estruturas e técnicas de programação e que seja capaz de escrever programas correctos.

É certo que a aprendizagem de novos conceitos em áreas como a programação requer a plena compreensão e assimilação dos conteúdos anteriormente ensinados, o que leva a que as disciplinas introdutórias de programação sejam frequentemente frustrantes, tanto para os alunos como para os próprios professores, devido à abstracção dos conceitos e pelo facto de muitas vezes o resultado não ser completamente visível. [FME2001].

Tudo isto leva muitas vezes ao aluno pensar que tais conceitos não trazem benefícios à sua aprendizagem.

Por este facto, o nosso interesse em aplicar problemas para serem resolvidos com o robot recai todo sobre Bases de Programação de 10.º ano, uma vez que será esta disciplina a base da aprendizagem da algoritmia e programação.

Para uma melhor preparação dos problemas que apresentamos no capítulo 6, identificamos os seguintes conceitos inerentes à disciplina:

- Noção de Algoritmo e Programa;
- Variáveis e Expressões;
- Estruturas de Decisão;
- Estruturas de Repetição;
- Vectores e Matrizes.

## 4.4. Conclusão

---

Um dos nossos objectivos, estabelecidos para esta dissertação, é “Criar problemas para serem resolvidos, usando os robots, nas disciplinas de programação do ensino secundário”.

Neste capítulo apresentamos uma análise curricular de todas as disciplinas do Curso Tecnológico de Informática do ensino secundário, onde se ensina informática. Esta análise permitiu-nos filtrar em quais dessas disciplinas são ensinados conceitos e técnicas de programação.

Devido ao facto da aprendizagem de novos conceitos em áreas como a programação requerer a plena compreensão e assimilação dos conteúdos anteriormente ensinados, decidimos direccionar o nosso trabalho para dar uma resposta a esta necessidade, escolhendo a disciplina de Bases de Programação 10.º ano para aplicar os problemas a serem resolvidos através dos robots.

Para suportar esta decisão, apresentamos na secção 5.2 do capítulo 5 uma plataforma que permitirá aos alunos programar os robots usando a linguagem Pascal, sugerida pelo Ministério como linguagem da disciplina, e no capítulo 6 um conjunto de problemas que abrangem todos os conceitos identificados na secção 4.3 deste capítulo.

---

## 5. Novas Plataformas de Programação

---

*“It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.”*

**Charles Darwin**

Na sequência do estudo efectuado sobre o kit da Lego® Mindstorms™ e das ferramentas existentes, apresentamos duas novas plataformas de programação para programar os robots. Na secção 5.1 apresentamos a plataforma RCX + Prolog, uma plataforma para programar os robots com a linguagem Prolog nas aulas de Inteligência Artificial, do Curso Engenharia Informática. Na secção 5.2 apresentamos a plataforma RCX+Pascal, uma plataforma para ser usada nas aulas de Bases de Programação do 10.º ano do ensino secundário. Por fim, na secção 5.3 apresentamos algumas conclusões do trabalho realizado.

### 5.1. RCX+Prolog

---

O RCX+PROLOG – Uma plataforma para usar os robots da Lego® Mindstorms™ nas aulas de Inteligência Artificial. Tem como objectivos:

- Trazer às aulas resultados concretos de forma a motivar os alunos para o desenvolvimento das suas tarefas e consequentemente melhorar a sua aprendizagem;

- Aliar à utilização dos robots, para a resolução de problemas, a utilização da linguagem de Programação PROLOG.

### 5.1.1. Motivação para a construção da plataforma

Definir Inteligência Artificial (IA) é uma tarefa difícil. Nos últimos anos surgiram quatro definições de IA, como mostra a tabela seguinte. [RN2003]

Sistemas que pensam como os Humanos	Sistemas que pensam racionalmente
<p>“O esforço para fazer com que os computadores pensem... máquinas com inteligência em todos os sentidos.” (Haugeland, 1985)</p> <p>“[A automatização de] actividades que associamos com o raciocínio humano, tais como a tomada de decisão, resolver de problemas, aprender...”(Bellman, 1978)</p>	<p>“O estudo de faculdades mentais através do uso de modelos computacionais” (Chamiak and McDermott, 1985)</p> <p>“O estudo da computação que permite perceber, raciocinar e agir.” (Winston, 1992)</p>
Sistemas que agem como os Humanos	Sistemas que agem racionalmente
<p>“A arte de criar máquinas que executam funções que requerem a inteligência quando executadas por pessoas.” (Kurzweil 1990)</p> <p>“O estudo de como é que os computadores fazem as coisas, no momento mas, nas quais as pessoas são melhores”. (Rich and Knight, 1991)</p>	<p>“Inteligência Computacional é o estudo da construção de agentes inteligentes.” (Poole et al., 1998)</p> <p>“IA... está interessada no comportamento inteligente dos artefactos.” (Nilsson, 1998)</p>
Algumas definições de Inteligência Artificial, organizadas em quatro categorias	

Tabela 5 – Definições de IA [RN2003]

Em qualquer um dos casos, é evidente que a IA pode tornar-se um conceito difícil de ensinar. Nesta perspectiva, o uso de robots pode trazer vantagens à aprendizagem da Inteligência Artificial, nomeadamente: [PS2004]

- “Os alunos estão habituados a lidar com robots no seu dia-a-dia nas tarefas mais comuns. Deste modo é mais fácil dar exemplos de robots bastante sofisticados que comunicam ideias de forma acessível.”

- “Para os alunos, conseguir que um robot tenha um determinado comportamento é uma tarefa motivante. Usar robots torna-se assim uma forma de capturar e manter o interesse no assunto.”

Contudo, temos de distinguir entre “ensinar robótica” e “ensinar com robots”. No primeiro caso, os robots são o objecto de estudo, enquanto que no segundo os robots são os mediadores para a aprendizagem de um determinado assunto.

No nosso caso, estamos interessados em usar os robots nas aulas de IA porque podem ser facilmente associados a Agentes. [FG1997]

A metáfora de agentes inteligentes foi fortemente usada em IA durante a última década. Um agente pode ser definido como:

- [RN2003] "Um agente é uma entidade que possa compreender o meio ambiente através de sensores e actuar sobre esse meio através de efectuadores”.
- [W2002] “Um agente é um sistema de computador capaz de acções autónomas num determinado contexto, de forma cumprir os seus objectivos”.

Perante estas definições é fácil perceber que o RCX pode ser visto como um agente, ilustrado na figura seguinte.

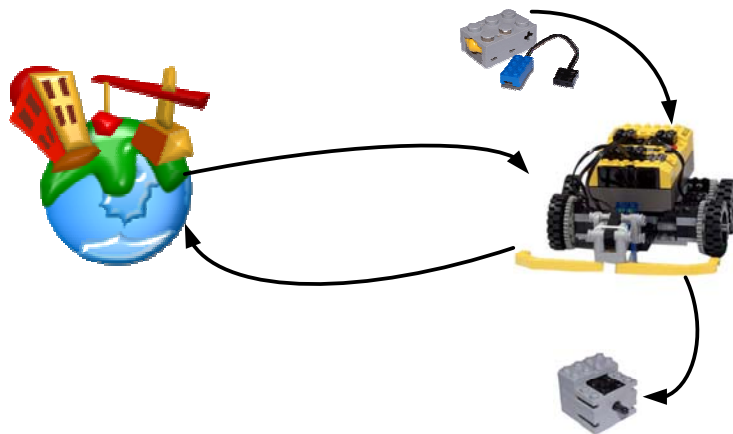


Figura 13 – O RCX como um Agente

Contudo, se queremos vê-lo como um agente inteligente temos de programá-lo de forma a “cumprir os seus objectivos”. Para tal, precisamos de um agente baseado em conhecimento. Um agente deste tipo inclui uma base de conhecimento acerca de um domínio e uma interface (programa de computador) constituída por regras para

processar o conhecimento e resolver os problemas relativos ao domínio. Resumindo, podemos usar a definição de Kowalsky [K1979]:

***Algoritmo = Logica + Controlo***

O Prolog (PROgramming in LOGic) foi criado em 1972 por Alain Colmerauer e Philippe Roussel, baseado na interpretação procedimental das cláusulas de Horn de Kowalski. O Prolog é usado para representar o conhecimento e para desenvolver soluções de problemas através da dedução lógica.

O Prolog é a linguagem mais usada ao longo dos tempos para programar na IA. Infelizmente para nós, não é possível programar directamente o RCX com a linguagem Prolog.

Perante tal situação, e tendo em conta todas as questões apresentadas, a nossa ideia foi criar uma plataforma, freeware, que permitisse aos alunos da disciplina de IA programar em Prolog o comportamento do Robot.

Como antecedente ao nosso trabalho, tivemos em conta a plataforma Legolog, criada por Levesque & Pagnucco, cujo objectivo era implementar cálculo situacional. Escrita em Prolog foi a nossa principal fonte de inspiração principalmente na comunicação Prolog – RCX. [LP2000] Contudo, existem três grandes diferenças em relação ao nosso trabalho:

1. Fizemos um protocolo de comunicação mais complexo, de forma a poder transmitir ao robot instruções com parâmetros. A ideia é aproximar estas instruções o máximo possível dos conceitos de blocos de programação (Avançar, Recuar, Virar à direita, entre outros) existentes na plataforma RIS. Por esta razão, a torre de infravermelhos comunica três parâmetros ao robot enquanto que o Legolog apenas transmitia um.
2. Desenvolvemos a plataforma para o sistema operativo Windows®. O Legolog funciona apenas sobre Linux.
3. O Legolog era orientado ao cálculo situacional. A nossa solução é Prolog “puro”.



### 5.1.2. Funcionamento da Plataforma

Tendo por base o Legolog, surgiu então o RCX+PROLOG: uma ferramenta para usar os robots da Lego® Mindstorms™ nas aulas de Inteligência Artificial. Esta ferramenta foi concebida para ser executada sobre o sistema operativo Windows, usando a torre de infravermelho ligada à porta USB, e a sua arquitectura está representada na figura seguinte.

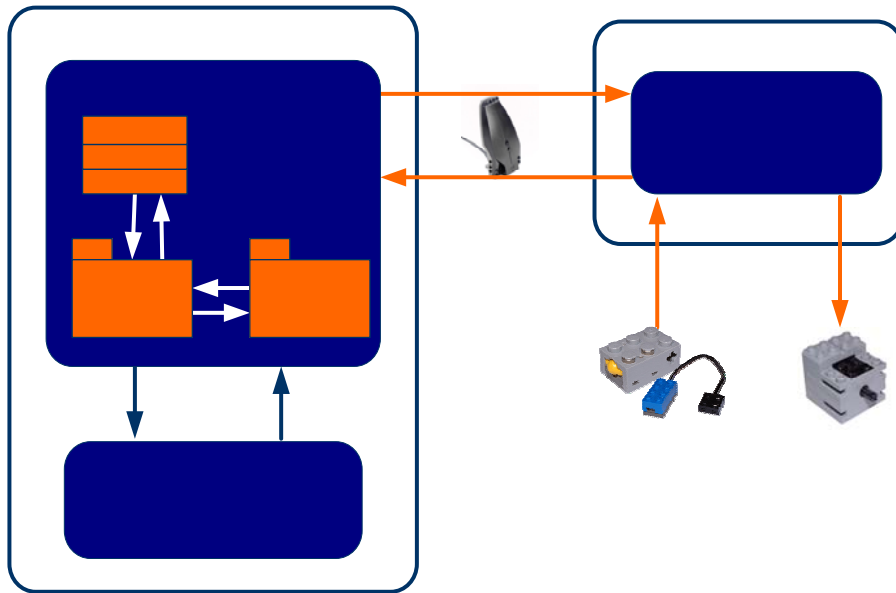


Figura 14 - Arquitectura do RCX+PROLOG

Computador

A comunicação entre o programa Prolog e o robot é efectuada pelo procedimento que se segue:

1. O programa Prolog é executado no Eclipse Control;
2. O programa envia à interface Java um código;
3. Esse código é retransmitido ao robot via torre de infravermelhos USB;
4. O robot recebe o código e confirma a recepção à interface Java;
5. Por fim reenvia a confirmação ao programa Prolog.

rcxjava

Com.

API

O código enviado pelo programa Prolog é constituído por um conjunto de três valores. O primeiro representa a acção (Avançar, Recuar, Virar até valor luz, etc.), o segundo e o terceiro valores são os parâmetros da acção, por exemplo, segundos e centésimas de segundos. Se não for necessário parâmetro para a acção o valor enviado é o 255.

A comunicação entre o computador e o robot está de uma forma resumida ilustrada na figura seguinte.

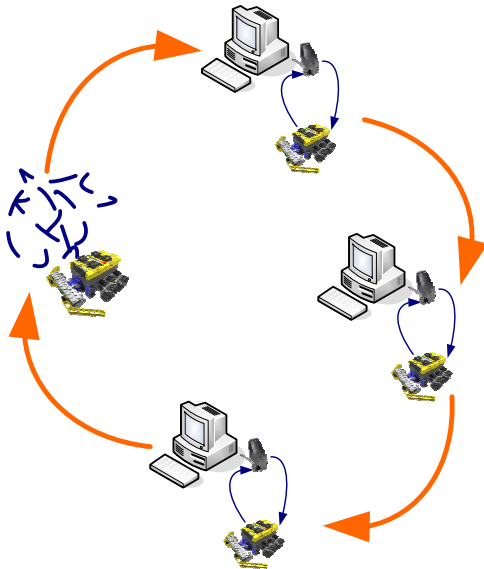


Figura 15 - Funcionamento do RCX+PROLOG

Uma vez que o PROLOG e o NQC trocam valores entre si, é necessário que ambas as partes comuniquem na mesma linguagem para que possam interpretar o significado dos valores emitidos. A Tabela seguinte mostra parcialmente, a estrutura do protocolo de comunicação criado (poderá consultar mais informação acerca do protocolo no anexo C).

4

Cód.	Acção	Parâmetro 1	Parâmetro 2
1	Avançar	Seg. [0-254]	Cent.[0-99]
2	Avançar Apenas Motor A	Seg. [0-254]	Cent.[0-99]
3	Avançar Apenas Motor B	Seg. [0-254]	Cent.[0-99]
4	Avançar Apenas Motor C	Seg. [0-254]	Cent.[0-99]
5	Avançar até Toque Sensor 1	-	-
6	Avançar até Toque Sensor 3	-	-
7	Avançar até Toque Sensor 1, 3	-	-
8	Avançar até Sensor de Luz	N.º Min [1-100]	N.º Máx.[1-100]
9	Avançar até Sensor de Luz, sensor de toque 1,3	N.º Min [1-100]	N.º Máx.[1-100]
10	Recuar	Seg. [0-254]	Cent.[0-99]
11	Recuar Apenas Motor A	Seg. [0-254]	Cent.[0-99]
12	Recuar Apenas Motor B	Seg. [0-254]	Cent.[0-99]
13	Recuar Apenas Motor C	Seg. [0-254]	Cent.[0-99]
14	Recuar até Toque Sensor 1	-	-
15	Recuar até Toque Sensor 3	-	-

OK

...

Tabela 6 – Estrutura do Protocolo de Comunicação do RCX+PROLOG

### 5.1.3. Módulos da Plataforma

A plataforma RCX+PROLOG está dividida em três módulos, nomeadamente o Módulo PROLOG, o Módulo de Comunicação e o Módulo NQC, que apresentamos de seguida.

#### Módulo PROLOG

Todo o “raciocínio” do robot é efectuado no computador através de um programa escrito em Prolog e compilado pelo Eclipse Prolog.

A tabela seguinte mostra um excerto do código implementado:

```
avancar(Segundos,Centesimas,Res):-  
    100>= Centesimas,  
    send_action(1,Segundos,Centesimas,Res).  
  
%send_action(+Num,+Par1,+Par2,-Res).  
mandar_accao_composta(Num,Par1,Par2,Res):-  
    rcxToStream(Rcx),  
    write_exdr(Rcx, Number),  
    flush(Rcx),  
    getRcxMess(_),  
    rcxToStream(Rcx),  
    write_exdr(Rcx,Par1),  
    flush(Rcx),  
    getRcxMess(_),  
    rcxToStream(Rcx),  
    write_exdr(Rcx,Par2),  
    flush(Rcx),  
    getRcxMess(Answer).
```

Tabela 7 – Excerto do código Prolog

Então, para que o robot tome decisões e execute acções, os alunos apenas precisam de compor o programa principal, usando (podem modificar se eventualmente for necessário) os “Blocos”/predicados do módulo de PROLOG.

#### Módulo de Comunicação

A comunicação é feita através de um conjunto de classes escritas em Java que irão servir de elo de ligação entre o Eclipse Prolog e o RCX.

Por um lado temos o RCX JAVA API [RJA], um pacote que providência uma API de alto nível que comunica com o RCX via torre de infravermelhos, ligada tanto na porta série ou na porta USB, e que gere todo o protocolo de comunicação, permitindo assim uma interface simples para controlar o Robot.

Por outro lado temos o Java Communication API [JCA], que serve de suporte de comunicação à API anterior. Esta API facilita o desenvolvimento de plataformas de comunicação independentes, como por exemplo Smart Cards, fax, modems e claro Robots.

Finalmente a classe Control.java, adaptada da classe escrita por Josh Singer – 2001, que serve de interface entre o Eclipse <-> RCX JAVA API <-> RCX. Esta Classe permite o envio de valores dos predicados escritos no programa em PROLOG para o RCX, bem como as respostas do RCX para o PROLOG.

### Módulo NQC

O programa escrito em NQC, executado no RCX, tem como responsabilidade receber números, enviar uma resposta “OK” e, consoante a acção e respectivos parâmetros, executar determinado comportamento no robot.

A tabela seguinte apresenta um excerto de código, que mostra como o RCX recebe e envia os valores, e determina qual a acção a executar.

```
task main()
{
    int accao, parametro1, parametro2;
    SetTxPower(TX_POWER_HI);
    ClearMessage();
    inicializa();

    while(true) {
        recebenums(accao, parametro1, parametro2);
        startaccao(accao, parametro1, parametro2);
    }
}
...
void Avancar(int parametro1, int parametro2)
{
    SetPower(ESQ_MOTOR + DIR_MOTOR, MOVE_SPEED);
    OnFwd(ESQ_MOTOR + DIR_MOTOR);
    Wait((parametro1*100)+parametro2);
    Off(ESQ_MOTOR + DIR_MOTOR);
}...
```

Tabela 8 – Excerto do Código NQC

## 5.2. RCX+Pascal

---

O RCX+Pascal – Uma plataforma para usar os robots da Lego® Mindstorms™ nas aulas de Bases de Programação do 10.º ano do Curso Tecnológico de Informática do Ensino Secundário. Tal como a plataforma anterior, tem como propósito dois factores:

- Reunir um conjunto de tecnologias existentes e criar alguns componentes, de forma a otimizar o processo de programação dos robots, usando a linguagem Pascal, no contexto da disciplina em questão;
- Trazer às aulas resultados concretos de forma a motivar os alunos para o desenvolvimento das suas tarefas e consequentemente melhorar a sua aprendizagem.

### 5.2.1. Motivação para a construção da plataforma

A Informática é uma área muito diversificada e, se nos focarmos no ensino da programação, torna-se numa área exigente. Muitas vezes existem conceitos fundamentais que são ensinados na sala de aula em relativamente pouco tempo. Além disso, os métodos tradicionais de ensino tendem a favorecer as abstrações, o que, frequentemente, faz com que os alunos tenham dificuldade em compreender o comportamento dos programas informáticos e reconhecer a relevância das tarefas que lhes são propostas. [AFF2006]

Pelas razões acima apresentadas propomos esta plataforma para usar o robot como mediador entre o aluno e os conceitos de programação (tal como na plataforma anterior, pretende-se “ensinar com robots”), de forma a motivar e tornar os conteúdos da programação mais pertinentes para os alunos e consequentemente contribuir para a melhoria do processo de aprendizagem.

Como base para o nosso trabalho tivemos em conta a ferramenta BricxCC, pelo facto de suportar a API de Pascal, e o firmware alternativo BrickOS, por suportar e executar programas escritos em linguagem Pascal.

Contudo, existe uma mais valia entre a simples integração destas tecnologias e o nosso trabalho. Desenvolvemos uma biblioteca, denominada de BIGBLOCKS, para aproximar a filosofia de programação do robot aos conceitos de blocos de programação (Avançar, Recuar, Virar à direita, entre outros) existentes na plataforma RIS.

### 5.2.2. Funcionamento da Plataforma

A plataforma é totalmente freeware e corre sobre o sistema operativo Windows, usando a torre de infravermelhos ligada à porta USB. A sua arquitectura está representada na figura seguinte:

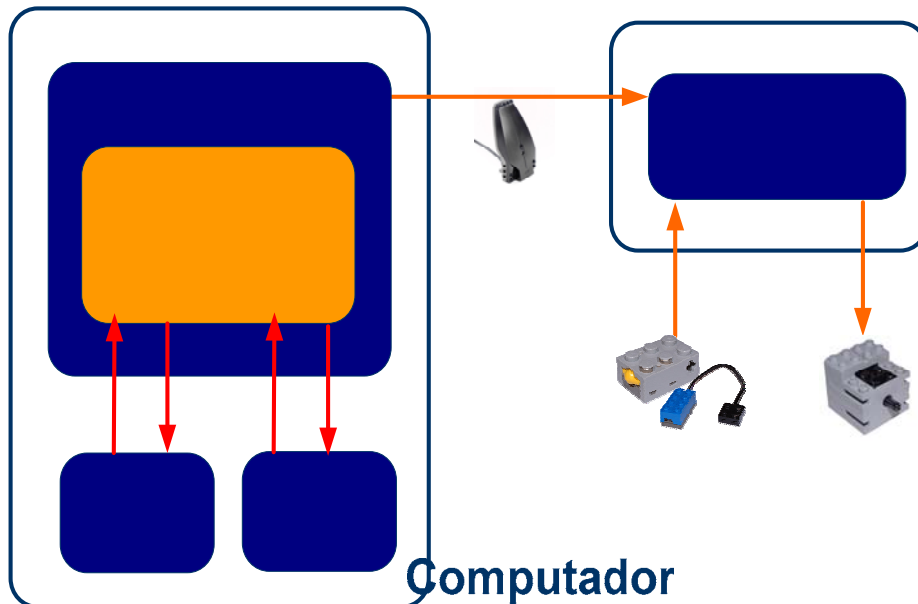


Figura 16 - Arquitectura do RCX+Pascal

De uma forma geral, o funcionamento desta plataforma é muito simples e baseia-se no seguinte procedimento:

- O programa em pascal é escrito na ferramenta BricxCC, recorrendo às funções da “API Pascal” e da biblioteca BIGBLOCKS;
- O código escrito é compilado no BricxCC;
- Depois de compilado é feito o download do programa para o RCX através da torre de infravermelhos;
- O programa é executado sobre o firmware BrickOS no Robot.

A figura seguinte ilustra o procedimento descrito anteriormente.

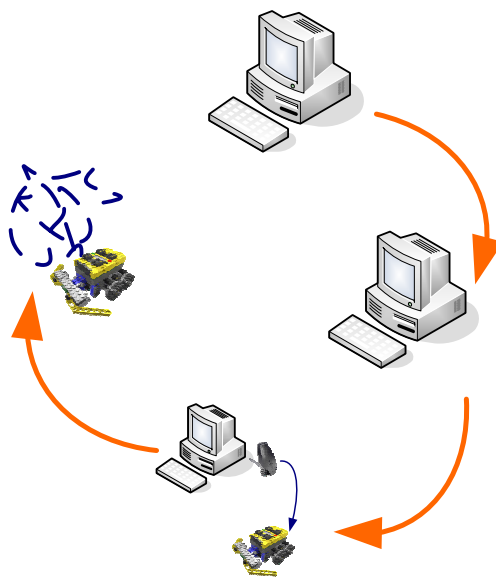


Figura 17 – Funcionamento do RCX + PASCAL

4

### 5.2.3. Módulos da Plataforma

A plataforma RCX+Pascal está dividida em três módulos, nomeadamente o BricxCC, a “API Pascal” e a Biblioteca BIGBLOCKS, que apresentamos de seguida.

#### Módulo – BricxCC

Este módulo, na sua essência, é a ferramenta BricxCC (apresentado no capítulo 3, secção 3.3.3) personalizada para suporta a programação do RCX em linguagem Pascal.

A vantagem da utilização do BricxCC deve-se ao facto do utilizador ter à sua disposição a possibilidade de poder compilar e depurar o programa, bem como poder usufruir de um leque de opções úteis como fazer download do programa para o RCX, instalar o firmware no RCX e até comandar o RCX remotamente.

No sítio <http://dme.uma.pt/droide/rcxpascal> apresentamos um pequeno guia de instalação e utilização desta ferramenta.

#### Módulo - “API Pascal”

A API de pascal tem um conjunto de funções definidas que permitem programar o robot usando a linguagem Pascal.

Estas funções permitem programar todos os componentes do robot, nomeadamente os motores, os sensores, o LCD, sons e envio de mensagens via infravermelhos. No anexo

A apresentamos um manual da API onde explicamos cada uma das funções e damos vários exemplos sobre cada uma delas.

A tabela seguinte mostra um programa escrito em Pascal usando as funções da API:

```
Program valorluz;

uses
  unistd, dsensor, dmotor;

begin
  ds_active(@SENSOR_2);

  if(LIGHT(SENSOR_2)=39) then
    begin
      motor_a_dir(mdFwd);
      motor_c_dir(mdFwd);
      motor_a_speed(255);
      motor_c_speed(255);
      sleep(2);
      motor_a_dir(mdOff);
      motor_c_dir(mdOff);

    end;
  end.
```

Tabela 9 – Exemplo de um programa escrito em Pascal

Analisando o código, temos um programa que compara o valor do sensor de luz ligado na porta 2 com um determinado valor. Se a intensidade da luz corresponder ao número 39 então o robot avança dois segundos.

### Módulo - “Biblioteca BIGBLOCKS”

Se olharmos para o exemplo anteriormente apresentado, para o robot avançar dois segundos precisamos do seguinte código:

```
motor_a_dir(mdFwd);
motor_c_dir(mdFwd);
motor_a_speed(255);
motor_c_speed(255);
sleep(2);
motor_a_dir(mdOff);
motor_c_dir(mdOff);
```

Ou seja, para um robot que se movimente com dois motores ligados às portas A e C poder avançar, precisamos de indicar um sentido aos dois motores, neste caso avançar. Seguidamente, é necessário indicar a potência do movimento, nesta situação a potência máxima. Depois, precisamos de indicar que as próximas instruções vão sofrer um atraso de 2 segundos e só depois desligamos os motores.



Este procedimento não é mais do que saber programar o robot, o que vai contra o nosso propósito de ensinar a programar usando o robot.

Perante esta situação, e à semelhança da plataforma anterior, o nosso objectivo era ter uma filosofia de programação o mais próxima ao da ferramenta RIS, ou seja, quando for necessário avançar, a única preocupação seria indicar ao robot que iria avançar num determinado tempo. Criamos a biblioteca BIGBLOCKS onde estão implementados os blocos de programação (Avançar, Recuar, Virar à direita, entre outros) existentes na plataforma RIS.

A tabela seguinte resume as funções implementadas na biblioteca BIGBLOCKS.

Acção	Função	Parâmetro 1	Parâmetro 2
Avançar Motor A e C	avancaAC(seg:integer; dseg:integer)	Segundos	Décimas S.
Avançar Motor A	avancaA(seg:integer; dseg:integer)	Segundos	Décimas S.
Avançar Motor B	avancaB(seg:integer; dseg:integer)	Segundos	Décimas S.
Avançar Motor C	avancaC(seg:integer; dseg:integer)	Segundos	Décimas S.
Recuar Motor A e C	recuaAC(seg:integer; dseg:integer)	Segundos	Décimas S.
Recuar Motor A	recuaA(seg:integer; dseg:integer)	Segundos	Décimas S.
Recuar Motor B	recuaB(seg:integer; dseg:integer)	Segundos	Décimas S.
Recuar Motor C	recuaC(seg:integer; dseg:integer)	Segundos	Décimas S.
Virar à Direita	virarDireita(seg:integer; dseg:integer)	Segundos	Décimas S.
Virar à esquerda	virarEsquerda(seg:integer; dseg:integer)	Segundos	Décimas S.
Rodar à Direita	rodarDireita(seg:integer; dseg:integer)	Segundos	Décimas S.
Rodar à esquerda	rodarEsquerda(seg:integer; dseg:integer)	Segundos	Décimas S.

Tabela 10 – Funções implementadas na biblioteca BIGBLOCKS

## 5.3. Conclusão

Apresentamos neste capítulo duas plataformas que permitem ensinar a programar usando os robots. A primeira, o RCX+PROLOG, para ser usada nas aulas de Inteligência Artificial e a segunda, o RCX+Pascal, para ser usada nas aulas de Bases de Programação do 10.º ano.

Em relação à plataforma RCX+PROLOG queremos salientar duas vantagens da sua utilização. A primeira delas diz respeito ao facto de ser uma solução generalista, porque é completamente independente do tipo de robot que seja construído. Embora os programas escritos em Prolog e em NQC considerem dois motores nas portas A e C, um sensor de luz na porta 2 e dois sensores de toque nas portas 1 e 3, eles são relativamente

fáceis de modificar para outra distribuição pelo facto de serem “open source” e estarem devidamente comentados para este tipo de situação.

A segunda vantagem advém do facto da plataforma ter a mesma filosofia de funcionamento da ferramenta nativa RIS (documentada no capítulo 3), onde cada acção é um bloco com parâmetros associados. No nosso caso um predicado Prolog corresponde a uma acção.

Em relação à plataforma RCX+Pascal acreditamos que é uma mais valia para o ensino da programação, principalmente porque será aplicada numa fase de introdução aos conceitos de programação o que permitirá uma melhor visualização dos conceitos.

Outra vantagem desta plataforma, tal como a anterior, é ter a mesma filosofia de funcionamento da ferramenta nativa RIS, o que cria um padrão na programação do robot.

No capítulo 7 apresentamos algumas das experiências resultantes da aplicação destas ferramentas no seu contexto real.

---

## 6. Problemas

---

*“Each problem that I solved became a rule which served afterwards to solve other problems.”*

**Rene Descartes**

Os problemas que apresentamos de seguida, agrupados em actividades, destinam-se à disciplina de Bases de Programação do 10.º ano de escolaridade, do curso Tecnológico de Informática.

A linguagem de programação usada para resolver os vários problemas é a Pascal, de acordo com o programa oficial da disciplina homologado pelo Ministério de Educação.

A ferramenta de programação adoptada é a plataforma RCX+Pascal apresentada no capítulo 5.

Para programar os robots os alunos terão de recorrer às funções definidas na “API Pascal” e na biblioteca BIGBLOCKS (explicado na secção 5.2.3 do capítulo anterior).

Os modelos dos robots recomendados para a resolução dos problemas são o “O Tanque” e o “O Todo - Terreno”, ilustrados nas figuras seguintes. A designação dos robots foi adoptada por nós, pelo que poderão ser encontrados, estes modelos de robots, com outros nomes.

Para a construção destes robots, podem ser consultadas as instruções em <http://dme.uma.pt/droide/rcxpascal> ou o manual de instruções disponível no kit Lego® Mindstorms™.



Figura 18 – “O Tanque”



Figura 19 – “O Todo - Terreno”

As actividades apresentadas estão divididas em duas partes. A primeira parte é uma proposta de trabalho para o aluno, enquanto que a segunda é composta por sugestões metodológicas para o professor.

Por questões de visualização e apresentação decidimos que seria melhor apresentar as actividades com o formato final. A seguir explicamos como foram organizados os conteúdos em cada uma das partes.

### Primeira parte - Aluno

A figura ilustra o formato e organização da actividade no que se refere à parte do aluno. É constituída por duas partes assinaladas na imagem por 1 e 2.

A área assinalada por 1 foi reservada para um título da actividade, que reflecte o assunto abordado. Na outra área, assinalada por 2, encontram-se as questões/problemas que compõe a actividade.

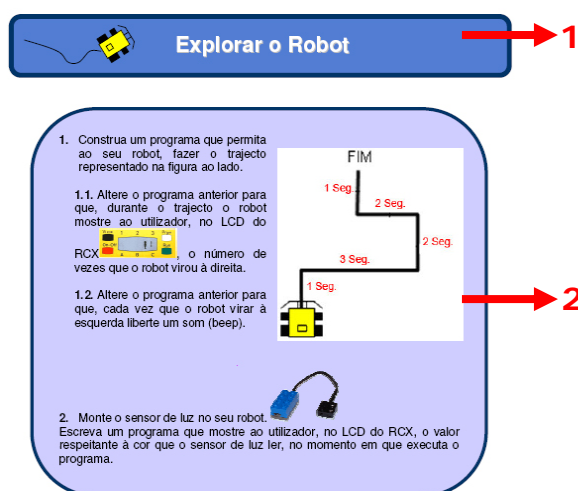


Figura 20 – Proposta de trabalho para o aluno

## Segunda parte - Professor

A segunda é constituída por oito áreas ilustrada na figura seguinte.

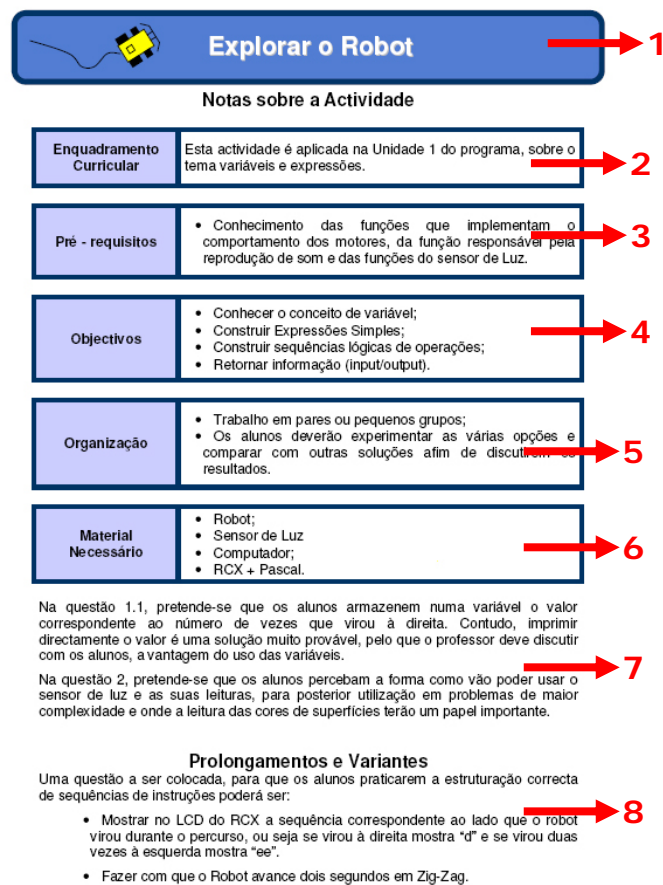


Figura 21 - Sugestões metodológicas para o professor

A seguir descrevemos cada uma das áreas:

- 1 - Tal como na parte dos alunos, representa o título da actividade;
- 2 - Área reservada ao enquadramento da actividade no programa da disciplina;
- 3 - Pré-requisitos necessários para os alunos poderem resolver os problemas da actividade;
- 4 - Objectivos que pretendemos que os alunos atinjam com a resolução da actividade;
- 5 - Sugestão de como o professor pode organizar o trabalho dos alunos na sala de aula;
- 6 - Material necessário para a execução da actividade;
- 7 - Contextualização para esclarecer o professor, do que se pretende com cada um dos problemas que compõem a actividade;

8 - Sugestões de problemas alternativos, quando necessários, que o professor pode colocar aos alunos ao longo ou depois de concluída a actividade.

A tabela seguinte apresenta o enquadramento global das várias actividades nos conteúdos da disciplina de Bases de Programação 10.º ano.

<b>Actividade</b>	<b>Tema</b>	<b>Título</b>	<b>Unidade</b>
Actividade 1	Introdução à Linguagem Pascal	Os Primeiros Passos	Unidade 1 – Noções Básicas
Actividade 2	Variáveis e Expressões	Explorar o Robot	Unidade 1 – Noções Básicas
Actividade 3	Programação Estruturada – Estruturas de Decisão	Vamos Decidir...	Unidade 2 – Algoritmia e Programação Estruturada
Actividade 4	Programação Estruturada – Estruturas de Repetição	Repete Robot, Repete...	Unidade 2 – Algoritmia e Programação Estruturada
Actividade 5	Estruturas de Repetição + Estruturas de Decisão	Se o Robot Repete...	Unidade 2 – Algoritmia e Programação Estruturada
Actividade 6	Vectores e Matrizes	Guardar Informação	Unidade 3 – Estruturas de Dados

Tabela 11 – Distribuição dos problemas pelos conteúdos programáticos

## 6.1. Actividade 1 - Introdução à Linguagem Pascal

---



### Os Primeiros Passos

**Programe o seu robot de forma a:**

1. Avançar durante um segundo.
2. Recuar durante um segundo.
3. Virar à esquerda.
4. Virar à direita.

**4.1.** Que alterações teria de fazer ao programa, para que agora o robot girasse à direita. Discuta com os colegas uma possível solução e teste-a no robot.

5. Libertar um som (beep).



## Os Primeiros Passos

### Notas sobre a Actividade

<b>Enquadramento Curricular</b>	Esta actividade é aplicada na Unidade 1 do programa, como introdução à Programação em Pascal.
<b>Pré - requisitos</b>	<ul style="list-style-type: none"><li>• Conhecimento do ambiente da Plataforma RCX + Pascal;</li><li>• Conhecimento do funcionamento do RCX.</li></ul>
<b>Objectivos</b>	<ul style="list-style-type: none"><li>• Introduzir a Linguagem de Programação Pascal;</li><li>• Conhecer termos e conceitos básicos relacionados com a programação do RCX em Pascal;</li><li>• Estruturar pequenos programas em Pascal.</li></ul>
<b>Organização</b>	<ul style="list-style-type: none"><li>• Trabalho em pares ou individual;</li><li>• Os alunos deverão experimentar as várias opções e comparar com outras soluções afim de discutirem os resultados.</li></ul>
<b>Material Necessário</b>	<ul style="list-style-type: none"><li>• Robot;</li><li>• Computador;</li><li>• RCX + Pascal.</li></ul>

Os alunos usam os robots construídos em aulas anteriores e, caso os robots sejam diferentes, pede-se a atenção do professor para o facto da velocidade dos robots e o tempo de execução dos programas serem diferentes.

Nesta actividade os alunos vão estruturar pequenos programas em Linguagem Pascal, recorrendo a funções específicas dos robots. No início da aula o professor deve expor aos alunos essas mesmas funções, embora sem grandes pormenores. Pretende-se que os alunos testem várias opções e discutam entre si, de forma a perceberem e consolidarem a estruturação correcta dos programas.

### Prolongamentos e Variantes

Os alunos podem testar os vários programas com outros tempos e perceber o que acontece. O professor pode pedir sempre que possível comparações com situações do dia-a-dia, bem como exemplos de aplicação destas pequenas operações.




## 6.2. Actividade 2 - Variáveis e Expressões



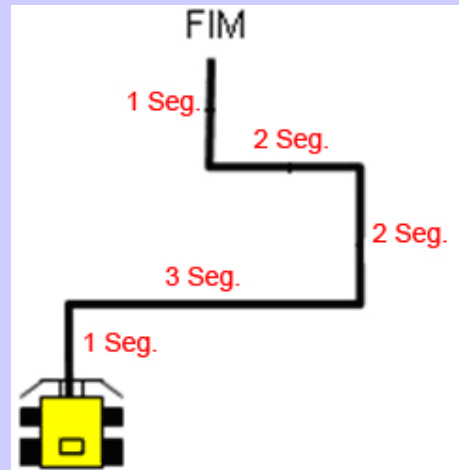
### Explorar o Robot

1. Construa um programa que permita ao seu robot fazer o trajecto representado na figura ao lado.

1.1. Altere o programa anterior para que, durante o trajecto, o robot mostre ao utilizador, no LCD do

RCX , o número de vezes que o robot virou à direita.

1.2. Altere o programa anterior para que, cada vez que o robot virar à esquerda, liberte um som (beep).



2. Monte o sensor de luz no seu robot. Escreva um programa que mostre ao utilizador, no LCD do RCX, o valor respeitante à cor que o sensor de luz ler no momento em que executa o programa.



## Explorar o Robot

### Notas sobre a Actividade

<b>Enquadramento Curricular</b>	Esta actividade é aplicada na Unidade 1 do programa, sobre o tema variáveis e expressões.
<b>Pré - requisitos</b>	<ul style="list-style-type: none"><li>• Conhecimento das funções que implementam o comportamento dos motores, da função responsável pela reprodução de som e das funções do sensor de luz.</li></ul>
<b>Objectivos</b>	<ul style="list-style-type: none"><li>• Conhecer o conceito de variável;</li><li>• Construir expressões simples;</li><li>• Construir sequências lógicas de operações;</li><li>• Retornar informação (input/output).</li></ul>
<b>Organização</b>	<ul style="list-style-type: none"><li>• Trabalho em pares ou pequenos grupos;</li><li>• Os alunos deverão experimentar as várias opções e comparar com outras soluções afim de discutirem os resultados.</li></ul>
<b>Material Necessário</b>	<ul style="list-style-type: none"><li>• Robot;</li><li>• Sensor de luz;</li><li>• Computador;</li><li>• RCX + Pascal.</li></ul>

Na questão 1.1 pretende-se que os alunos armazenem numa variável o valor correspondente ao número de vezes que virou à direita. Contudo, imprimir directamente o valor é uma solução muito provável, pelo que o professor deve discutir com os alunos a vantagem do uso das variáveis.

Na questão 2 pretende-se que os alunos percebam a forma como vão poder usar o sensor de luz e as suas leituras, para posterior utilização em problemas de maior complexidade e onde a leitura das cores de superfícies terão um papel importante.

### Prolongamentos e Variantes

Uma questão a ser colocada para os alunos praticarem a estruturação correcta de sequências de instruções poderá ser:

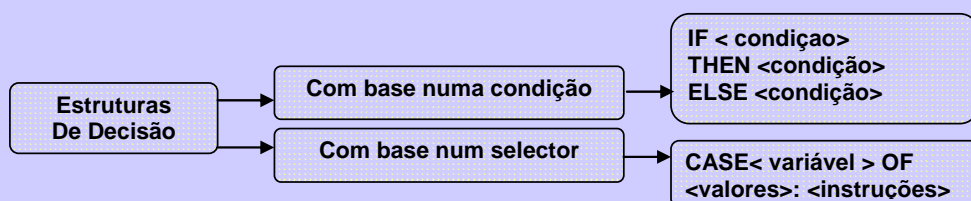
- Mostrar no LCD do RCX a sequência correspondente ao lado que o robot virou durante o percurso, ou seja, se virou à direita mostra “d” e se virou duas vezes à esquerda mostra “ee”.
- Fazer com que o Robot avance dois segundos em Zig-Zag.

## 6.3. Actividade 3 – Estruturas de Decisão



### Vamos Decidir...

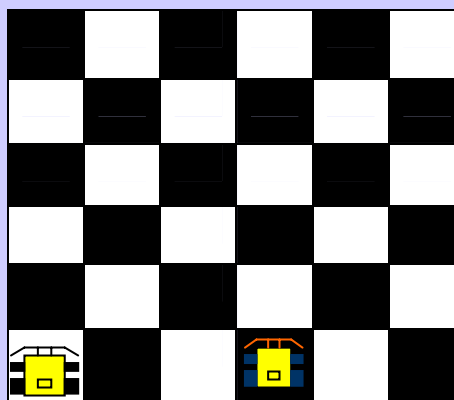
#### As Estruturas de Decisão



1. Considere como ponto de partida um dos quadrados do tabuleiro representado na imagem. Programe o seu robot para que:

1.1. Se estiver num quadrado preto volte à direita e termine a execução.

1.2. Se estiver num quadrado preto avance dois quadrados e termine a execução.



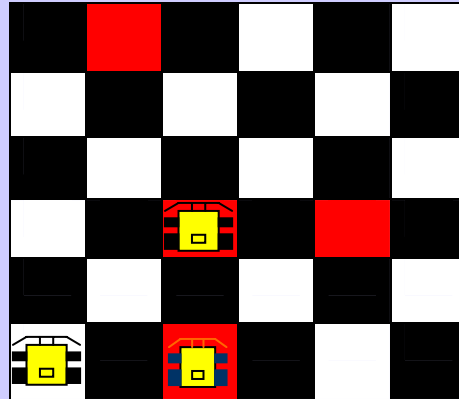
2. Faça um programa semelhante ao anterior, mas agora o robot só avança dois quadrados quando detectar qualquer cor excepto o preto.
3. Altere o programa anterior para que o robot avance para qualquer cor, excepto a cor preta e cor branca.
4. Monte no robot os dois sensores de toque.
  - 4.1. Programe o seu robot para reagir quando pressionado um dos sensores. Considere os seguintes comportamentos:
    - **Pressionar Sensor 1:** Recua durante dois segundos e pára;
    - **Pressionar Sensor 2:** Avança um segundo, volta à direita e pára.
  - 4.2. Tendo em conta que pressionar em simultâneo os dois sensores é uma situação muito provável, altere o programa anterior para quando tal situação acontecer o robot liberte um som (beep).

5. Considere como ponto de partida um dos quadrados do tabuleiro representado na imagem. Programe o seu robot para que:

- Se estiver num quadrado preto volte à direita e termine a execução.

- Se estiver num quadrado preto avance dois quadrados e termine a execução.

- Se estiver num quadrado vermelho avance um quadrado, liberte um som e termine a execução



6. Escreva o programa da questão 6, usando a estrutura de decisão "CASE...OF".



## Vamos Decidir...

### Notas sobre a Actividade

<b>Enquadramento Curricular</b>	Esta actividade é aplicada na Unidade 2 do programa, sobre o tema Estruturas de Decisão.
<b>Pré - requisitos</b>	<ul style="list-style-type: none"><li>• Conhecimento das funções que implementam o comportamento dos motores, da função responsável pela reprodução de som, das funções do sensor de luz e das funções do sensor de toque.</li></ul>
<b>Objectivos</b>	<ul style="list-style-type: none"><li>• Conhecer o conceito de estrutura de decisão;</li><li>• Aplicar estruturas de decisão na elaboração de algoritmos;</li><li>• Distinguir as diferentes estruturas de decisão;</li><li>• Distinguir as diferentes fases na resolução de um problema.</li></ul>
<b>Organização</b>	<ul style="list-style-type: none"><li>• Trabalho em pares ou pequenos grupos;</li><li>• Os alunos deverão experimentar as várias opções e comparar com outras soluções afim de discutirem os resultados.</li></ul>
<b>Material Necessário</b>	<ul style="list-style-type: none"><li>• Robot;</li><li>• Sensor de Luz;</li><li>• Sensor de Toque;</li><li>• Tabuleiro;</li><li>• Computador;</li><li>• RCX + Pascal.</li></ul>

Na questão 1 pretende-se que os alunos, perante dois cenários possíveis, ou seja o robot ou está num quadrado preto ou num quadrado branco, sejam capazes de aplicar uma estrutura de decisão que, dependendo de uma leitura inicial à superfície, vai determinar o comportamento do robot.

Na questão 2 pretende-se que os alunos percebam a forma como podem construir condições para serem avaliadas em situação de excepção.

Na questão 3 o objectivo é que os alunos sejam capazes de construir uma condição composta para a estrutura de decisão.

Na questão 4 o principal objectivo prende-se com o facto do aluno usar pela primeira vez o sensor de toque, pois em relação à solução do problema o raciocínio é análogo

aos exercícios anteriores. O professor pode aproveitar este exercício para consolidar os conhecimentos dos alunos.

Na questão 5 pretende-se que o aluno utilize a estrutura de decisão “IF...THEN...ELSE” encadeada para resolver a questão de ter três condições possíveis. Neste exercício o professor pode introduzir o conceito da estrutura de decisão “CASE...OF” e passar à resolução da questão 6.

Na questão 6 pretende-se que os alunos percebam a vantagem de usar o “CASE...OF” quando o número de condições aumenta.

### **Prolongamentos e Variantes**

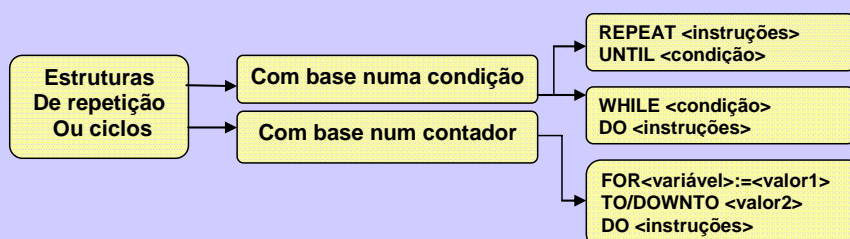
Aumentar o número de cores no tabuleiro, de forma a aumentar o número de condições para o robot, é um exercício útil para os alunos consolidarem os conhecimentos adquiridos.

## 6.4. Actividade 4 – Estruturas de Repetição



### Repete Robot, Repete...

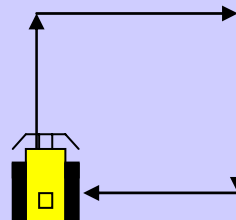
#### As Estruturas de Repetição



Utilizando a Estrutura de repetição “FOR...TO...DO”:

1. Faça um programa que permita ao seu robot repetir cinco vezes o seguinte comportamento: avançar um segundo, libertar um som e recuar um segundo.
2. Escreva um programa que permita ao robot realizar uma trajectória em forma de quadrado.

**Sugestão:** Poderá acoplar um marcador ao seu robot para que este desenhe o quadrado conforme realizar a trajectória.



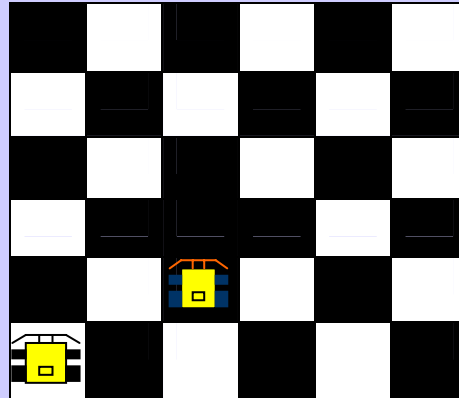
3. Altere o programa anterior para que o robot desenhe o quadrado 3 vezes.
4. Programe o seu robot para que repita cinco vezes o seguinte comportamento: avançar n segundos e recuar um segundo. Considere que n varia de [1-5] e aumenta uma unidade em cada iteração do ciclo.
5. Implemente as questões 1.1, 1.2 e 1.3, mas agora usando a estrutura de repetição “WHILE...DO”.

**6.** Considere um tabuleiro semelhante ao da figura.

O objectivo é fazer o robot avançar no tabuleiro até encontrar um quadrado de cor preta. Programe o seu robot:

**6.1.** Sem considerar o quadrado inicial (ponto de partida).

**6.2.** Considerando o quadrado inicial (ponto de partida).



**7.** Monte na parte da frente do seu robot os dois sensores de toque.

**7.1.** Programe o seu robot para avançar em “zig-zag”. O seu movimento será interrompido, ou seja o robot pára, quando um dos sensores for activado/pressionado.

**7.2.** Que alteração teria de fazer ao programa para que o robot parasse quando fossem pressionados os dois sensores em simultâneo.





## Repete Robot, Repete...

### Notas sobre a Actividade

<b>Enquadramento Curricular</b>	Esta actividade é aplicada na Unidade 2 do programa, sobre o tema Estruturas de Repetição.
<b>Pré - requisitos</b>	<ul style="list-style-type: none"><li>• Conhecimento das funções que implementam o comportamento dos motores, da função responsável pela reprodução de som, das funções do sensor de luz e das funções do sensor de toque.</li></ul>
<b>Objectivos</b>	<ul style="list-style-type: none"><li>• Conhecer o conceito de estrutura de repetição;</li><li>• Aplicar estruturas de repetição na elaboração de algoritmos;</li><li>• Distinguir as diferentes estruturas de repetição;</li><li>• Distinguir as diferentes fases na resolução de um problema.</li></ul>
<b>Organização</b>	<ul style="list-style-type: none"><li>• Trabalho em pares ou pequenos grupos;</li><li>• Os alunos deverão experimentar as várias opções e comparar com outras soluções afim de discutirem os resultados.</li></ul>
<b>Material Necessário</b>	<ul style="list-style-type: none"><li>• Robot;</li><li>• Sensor de luz;</li><li>• Sensor de toque;</li><li>• Computador;</li><li>• Tabuleiro;</li><li>• RCX + Pascal.</li></ul>

Na questão 1 pretende-se que os alunos percebam o conceito e utilidade de uma estrutura de repetição. O professor deve conduzir as soluções dos alunos de forma que eles percebam como devem de construir o ciclo e que instruções pertencem a esse ciclo.

Na questão 2 o objectivo é análogo ao da questão anterior. Pretende-se que os alunos consolidem o raciocínio da aplicação de estruturas de repetição em diferentes situações.

Na questão 3 o professor não deve explicar logo o conceito de ciclos encadeados. Deve antes deixar que os alunos exponham as suas soluções. Pretende-se com esta questão que os alunos percebam o conceito de ciclos encadeados.

Na questão 5 pretende-se que os alunos saibam estruturar os mesmos programas mas recorrendo à estrutura de repetição “WHILE...DO” e desta forma identificar as alterações necessárias aquando do seu uso.

Na questão 6 pretende-se que os alunos distingam qual das estruturas a aplicar é a mais adequada para a resolução de determinados problemas. O professor não deve indicar qual a estrutura a usar. Deve, pelo contrário, guiar o raciocínio do aluno, relembrando as características das estruturas de decisão estudadas.

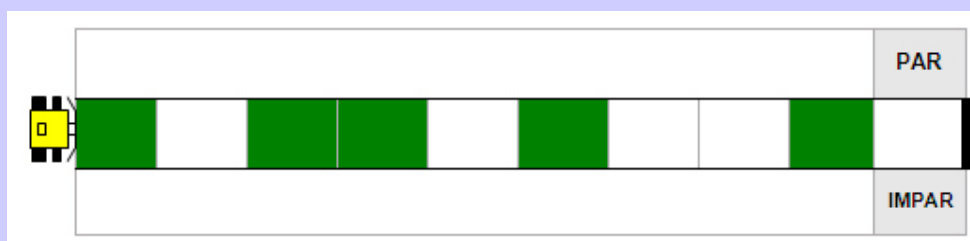
Na questão 7 pretende-se que aluno consolide o conceito de estruturas de repetição baseadas sobre uma condição e que sejam capazes de construir uma condição composta.

## 6.5. Actividade 5 - Estruturas de Repetição + Estruturas de Decisão



### Se o Robot Repete...

1. Considere a pista seguinte.

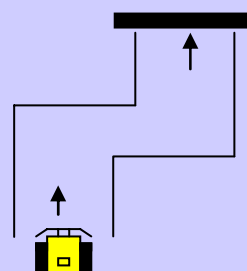


1.1. Programe o seu robot para que, depois de percorrer a pista, retorne ao utilizador (através do LCD do RCX) o número de quadrados verdes que encontrou no seu trajecto.

1.2. Altere o programa anterior para que o robot no fim do trajecto se desloque para o lado "PAR" ou "ÍMPAR", de acordo com o número de quadrados contados.

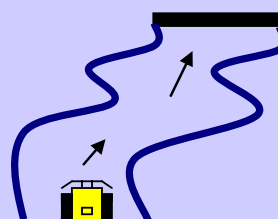
1.3. Altere a disposição e número de quadrados verdes da pista para testar outras possibilidades.

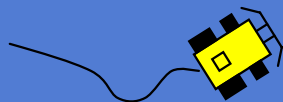
2. Programe o seu robot para realizar um trajecto semelhante ao da figura, evitando os obstáculos e/ou ajustando a sua trajectória sempre que necessário. O fim do trajecto é representado por uma faixa preta.



3. Escreva um programa para que o robot seja capaz de realizar um trajecto marcado por uma linha preta, ou seja, o robot deve ser capaz de seguir essa linha. O robot pára quando for activado um dos sensores de toque.

4. Escreva um programa para que o robot seja capaz de realizar um trajecto delineado por duas linhas azuis. O fim do trajecto é representado por uma faixa preta, como ilustra a figura.





## Se o Robot Repete...

### Notas sobre a Actividade

<b>Enquadramento Curricular</b>	Esta actividade é aplicada no fim da Unidade 2 do programa, sobre o tema Estruturas de Decisão + Repetição.
<b>Pré - requisitos</b>	<ul style="list-style-type: none"><li>• Conhecimento das funções que implementam o comportamento dos motores, da função responsável pela reprodução de som, das funções do sensor de luz e das funções do sensor de toque;</li><li>• Conhecimento das estruturas de decisão;</li><li>• Conhecimento das estruturas de repetição.</li></ul>
<b>Objectivos</b>	<ul style="list-style-type: none"><li>• Aplicar de forma conjugada as estruturas de decisão e repetição na elaboração de algoritmos;</li><li>• Desenvolver programas em Pascal, de forma estruturada;</li><li>• Decompor um problema nas suas partes.</li></ul>
<b>Organização</b>	<ul style="list-style-type: none"><li>• Trabalho em pares ou pequenos grupos;</li><li>• Os alunos deverão experimentar as várias opções e comparar com outras soluções afim de discutirem os resultados.</li></ul>
<b>Material Necessário</b>	<ul style="list-style-type: none"><li>• Robot;</li><li>• Sensor de luz;</li><li>• Sensor de toque;</li><li>• Computador;</li><li>• Pistas;</li><li>• RCX + Pascal.</li></ul>

Nesta altura, os alunos já têm alguma capacidade de programar e identificar soluções para diversos problemas. O objectivo comum a todos estes problemas é que o aluno seja capaz de construir programas para problemas mais elaborados, conjugando todos os conceitos adquiridos até ao momento.

Pretende-se desta forma que o aluno adquira a capacidade de decompor os problemas nas suas partes e que ao resolver essas partes obtenha a solução para o todo.

## 6.6. Actividade 6 - Vectores e Matrizes



### Guardar Informação

1. Considere a pista representada na figura seguinte. Tenha em conta que:

- A pista tem um tamanho fixo de 10 quadrados;
- Um quadrado de cor verde equivale a 3 valores;
- Um quadrado de cor preta equivale a 2 valores;
- Um quadrado de cor branca equivale a 1 valor.



1.1. Escreva um programa que guarde num vector os valores correspondentes a cada uma das cores dos quadrados da pista. No fim do percurso o robot deve de mostrar ao utilizador (através do LCD do RCX) o somatório desses valores.

1.2. Altere o programa anterior para que o robot retorne à média dos valores do vector.

2. Considere o seguinte cenário. Tenha em conta que:

- A cor preta equivale a 2 valores;
- A cor branca equivale a 1 valor;
- Cada vector é composto por 5 unidades que correspondem a 5 quadrados;



2.1. Programe o seu robot para que seja capaz de comparar dois vectores. Sempre que a cor do índice do Vector2 correspondente ao índice Vector1 for diferente, o robot deverá libertar um som (beep), para assinalar tal situação.

2.2. Faça um programa para que o robot retorne à média dos valores dos dois vectores.

3. Considere a pista da figura seguinte, que representa um byte, tendo em conta que:

- O vector é composto apenas pelos números do sistema binário;
- A cor preta equivale a 1 valores;
- A cor branca equivale a 0 valor.



**3.1.** Escreva um programa para que o seu robot, guarde num vector os valores correspondentes a cada posição da pista. Tendo em conta que o vector simboliza um byte, o robot deverá mostrar ao utilizador o valor decimal correspondente ao valor binário armazenado no vector.

**Nota:** Para a pista representada na figura, o valor a retornar seria 5.

**3.2.** Altere a disposição da pista e teste o robot com outras combinações.

4. Considere o tabuleiro da figura seguinte, que representa uma matriz (5X5). Tenha em conta que o robot deve partir da posição representada na figura.

**4.1.** Programe o robot para se deslocar para o índice (3,4) da matriz.

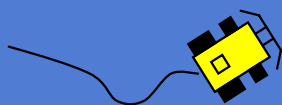
**4.2.** Escreva um programa para que o robot percorra o tabuleiro e guarde todos os valores numa matriz. Considere que:

- A cor verde equivale a 2 valores;
- A cor preta equivale a 1 valores;
- A cor branca equivale a 0 valores.

	1	2	3	4	5
1					
2					
3					
4					
5					

**4.3.** Altere o programa anterior, para que o robot mostre ao utilizador (através do LCD do RCX) o somatório dos elementos da diagonal.

**4.4.** Altere a disposição dos elementos do tabuleiro e teste o robot com outras combinações.



## Guardar Informação

### Notas sobre a Actividade

<b>Enquadramento Curricular</b>	Esta actividade é aplicada na Unidade 3 do programa, sobre o tema Vectores e Matrizes.
<b>Pré - requisitos</b>	<ul style="list-style-type: none"><li>• Conhecimento das funções que implementam o comportamento dos motores, da função responsável pela reprodução de som, das funções do sensor de luz e das funções do sensor de toque;</li><li>• Conhecimento das estruturas de decisão;</li><li>• Conhecimento das estruturas de repetição.</li></ul>
<b>Objectivos</b>	<ul style="list-style-type: none"><li>• Conhecer o conceito de estrutura de dados;</li><li>• Aplicar as estruturas de dados na elaboração de algoritmos em Pascal;</li><li>• Manipular valores das estruturas de dados;</li><li>• Decompor um problema nas suas partes.</li></ul>
<b>Organização</b>	<ul style="list-style-type: none"><li>• Trabalho em pares ou pequenos grupos;</li><li>• Os alunos deverão experimentar as várias opções e comparar com outras soluções afim de discutirem os resultados.</li></ul>
<b>Material Necessário</b>	<ul style="list-style-type: none"><li>• Robot;</li><li>• Sensor de luz;</li><li>• Sensor de toque;</li><li>• Computador;</li><li>• Tabuleiros;</li><li>• RCX + Pascal.</li></ul>

Na questão 1 pretende-se que os alunos percebam o conceito de vector e como podem implementá-los em Pascal. O professor deve explicar a forma como os alunos podem percorrer e aceder aos valores do vector. Pretende-se ainda com a resolução das alíneas que os alunos percebam que cada posição do vector pode ser vista como uma variável.

Na questão 2 pretende-se que os alunos manipulem valores de dois vectores diferentes. Pretende-se ainda que os alunos tenham a capacidade de resolver sozinhos a questão da transição de um vector para o outro. O professor deve ajudar os alunos nesta questão lançando dicas e apoiando cada uma das soluções encontradas.

Na questão 3 pretende-se que os alunos consolidem e pratiquem todas as questões inerentes aos vectores e manipulação dos seus dados.

Na questão 4 pretende-se que os alunos percebam o conceito de matriz, essencialmente na visão de vector bidimensional, ou seja, cada um dos seus elementos é acedido por dois índices. O professor deve explicar todos os conceitos relacionados com matrizes e utilizar o conceito de ciclos encadeados para explicar aos alunos como podem percorrer todos os elementos da matriz. Na alínea 3 pretende-se que o robot leia todos os valores, mas o cálculo do somatório depende da condição de considerar apenas os elementos da diagonal.

### **Prolongamentos e Variantes**

Como variante à questão 1 o professor pode pedir aos alunos para programar o robot, para que no fim do percurso o robot retorne o somatório dos valores das posições pares do vector.

Como variante à questão 4 o professor pode pedir aos alunos para programar o robot, para que retorne o somatório dos valores da diagonal inferior da matriz.



## 6.7. Conclusão

---

Acreditamos que a utilização dos robots nas aulas, de uma disciplina de introdução à programação, é uma mais valia para a compreensão dos conceitos. Os robots possibilitam resultados visíveis e concretos, motivando os alunos para a aprendizagem de conceitos mais avançados.

Apresentamos uma série de problemas, agrupados em várias actividades, que cobrem todos os conceitos da disciplina de Bases de Programação de 10.º ano. As actividades foram criadas a pensar na evolução do aluno e dos conceitos que deve de assimilar para resolver a actividade seguinte.

As actividades foram criadas em duas partes, uma como proposta de trabalho para o aluno e outra como sugestões metodológicas para o professor. Pretendemos assim disponibilizar apoio aos professores nesta nova perspectiva de ensino.

Tivemos o cuidado de criar uma actividade (Actividade 5), para consolidar e unir conceitos, no caso as estruturas de decisão e repetição. Pretende-se que nesta actividade o aluno compreenda de que forma, pode estruturar algoritmos mais complexos, usando todos os conceitos já estudados.

Finalmente julgamos que a aplicação destas actividades nas aulas de Bases de Programação 10.º ano, será uma mais valia no processo de ensino aprendizagem, contudo a sua aplicação e análise de resultados ficará como sugestão para continuidade deste trabalho.

---

## 7. Os Robots em Acção

---

*“Practice proves more than theory, in any case”*

**Abraham Lincoln**

Neste capítulo descrevemos algumas das nossas experiências da aplicação dos robots em conjunto com as plataformas apresentadas.

Na secção 7.1 descrevemos a acção de formação realizada na Escola Básica e Secundária Padre Manuel Álvares, para professores de informática e futuros profissionais de ensino (finalistas da Licenciatura de Ensino de Informática). Na secção 7.2 expomos os projectos, para serem resolvidos com os robots em conjunto com a plataforma RCX+PROLOG, aplicados na disciplina de Inteligência Artificial nos dois últimos anos lectivos. Por fim, em jeito de conclusão, apresentamos algumas reflexões sobre as experiências.

### 7.1. Acção de Formação

---

Esta acção decorreu no dia 26 de Maio de 2006, integrada no evento “I Workshop Tecnológico” da Escola Básica e Secundária Padre Manuel Álvares.

O público-alvo desta acção foi professores e futuros professores de informática (finalistas da Licenciatura em Ensino de Informática). Estiveram presentes 21 formandos de diferentes escolas da Região.

A acção de formação teve os seguintes objectivos:

- Apresentar o kit Lego® Mindstorms™ aos professores para que compreendessem o funcionamento dos robots, constatassem as suas

capacidades e limitações e se iniciassem no ambiente de programação da Robotics Invention System™ 2.0 e na linguagem de programação RCX code;

- Apresentar as plataformas RCX+PROLOG e RCX+Pascal e discutir o contexto e vantagens da sua aplicação;
- Apresentar uma proposta de actividade para ser aplicada nas aulas de Bases de Programação 10.º ano, com o intuito de mostrar como podem ser transformados os problemas tradicionais em problemas inovadores, para serem executados nos robots.

Os formandos trabalharam em grupos. A cada grupo foi disponibilizado um kit Lego® Mindstorms™, um manual de instruções da construção do robot e um portátil com todo o software necessário instalado.

A sessão decorreu em três partes distintas. Na primeira parte foi apresentado aos formandos o kit Lego® Mindstorms™, nomeadamente as peças mais importantes, como o RCX e os sensores de toque e de luz, seguido da construção de um robot a partir de um manual de construção.



Figura 22 - Construção do robot

Depois da construção foi dado a conhecer o software RIS e uma breve introdução da sua utilização, à qual se seguiu a proposta de realização de três exercícios simples.

O primeiro exercício resumia-se a fazer o robot avançar cinco segundos, abanar três vezes e, por fim, emitir um sinal sonoro.

O segundo propunha que construíssem um programa para o robot descrever uma trajectória em forma de quadrado.

No terceiro exercício foi pedido aos formandos que programassem os robots para percorrer um labirinto de forma aleatória.



Figura 23 – Execução do exercício 3

Para executar este exercício precisavam de adicionar ao robot um “pára-choques inteligente” com dois sensores de toque, fornecido por nós.

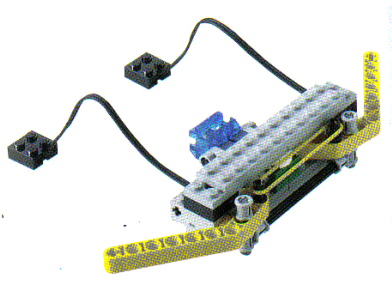


Figura 24 – “Pára-choques inteligente”

Na segunda parte apresentamos a plataforma RCX+PROLOG, demonstramos um exemplo da sua aplicação, neste caso particular o “Mundo de Wumpus” (documentado na secção 7.2. deste capítulo) e analisamos as vantagens da sua utilização.

A terceira parte ficou reservada à questão: Como usar os robots nas aulas de Bases de Programação?

Pretendíamos dar aos formandos presentes uma visão geral, de como poderiam ser usados os robots para ensinar a programar, quais as principais diferenças deste novo método de ensino e quais as necessidades para a sua aplicação.

Começamos por apresentar um exemplo prático de como transformar uma questão dita tradicional, numa questão para ser resolvida usando o robot, apresentado na tabela seguinte:

**Problema “Tradicional”:** Usando uma estrutura de repetição, escreva um programa que imprima no ecrã 5 vezes a palavra DROIDE, finalizando com a frase “FIM DO PROGRAMA”.

**Problema para o robot:** Usando uma estrutura de repetição, escreva um programa para que o robot seja capaz de avançar e recuar 5 vezes, finalizando com um som.

Tabela 12 – Transformação de um problema

Depois apresentamos a plataforma RCX + Pascal e todo o seu modo de funcionamento. Os formandos testaram a plataforma, já instalada nos seus computadores, resolvendo em conjunto dois problemas que cobriam situações simples da utilização de estruturas de repetição (ver anexo D).

## 7.2. Disciplina de Introdução à Inteligência Artificial

---

Introduzimos a plataforma RCX+PROLOG na disciplina de Inteligência Artificial do curso de Engenharia Informática da Universidade da Madeira no segundo semestre do ano lectivo de 2005/2006.

Os alunos têm de resolver um desafio proposto como projecto prático de avaliação usando os robots e a plataforma RCX+PROLOG.

No ano lectivo 2005/2006 o projecto escolhido foi “O Mundo de WUMPUS” e no ano lectivo 2006/2007 foi “O Problema dos Canibais e Missionários”.

A seguir descrevemos o contexto de cada um destes projectos. Mostramos também alguns dos resultados dos projectos realizados pelos alunos.

### 7.2.1. 2005/2006 – “O Mundo de Wumpus”

Como era a primeira vez que os alunos iriam usar a plataforma, decidimos usar um projecto bem conhecido, para evitar eventuais “surpresas”. Por esta razão escolhemos “O Mundo de Wumpus”, documentado no livro de Russell e Norvig [RN2003]. No entanto, modificamos ligeiramente o problema no sentido de que o agente não pode matar o Wumpus.

O projecto foi desenvolvido por grupos compostos por três alunos.

Russell e Norvig descrevem desta forma “O Mundo de Wumpus”:

*“O mundo de Wumpus é uma caverna constituída por divisões ligadas por passagens. Infelizmente algures na caverna está o Wumpus, um monstro que come qualquer um que entre na sua divisão. [...] Algumas divisões contêm buracos que “engolem”, qualquer um que vaguei por eles, à excepção do Wumpus, que é demasiado grande cair lá dentro. A única compensação de viver neste ambiente é a possibilidade de encontrar um cofre cheio de ouro.*

*O agente tem sensores, onde cada um percebe um bocado de informação:*

- *No quadrado que contem o Wumpus e nos quadrados directamente adjacentes (não diagonalmente) o agente perceberá um cheiro.*
- *Nos quadrados adjacentes a um poço, o agente perceberá uma brisa.*
- *No quadrado onde está o ouro, o agente perceberá um brilho.*
- *Quando um agente for contra uma parede, perceberá uma colisão.”*

Entretanto, os nossos robots não têm os sensores para perceber o cheiro, a brisa e o brilho. Por esta razão modificamos o tabuleiro, colocando marcas que representam a brisa, o cheiro e o brilho, que podem ser reconhecidas pelo sensor de luz. A figura que se segue representa as alterações efectuadas.

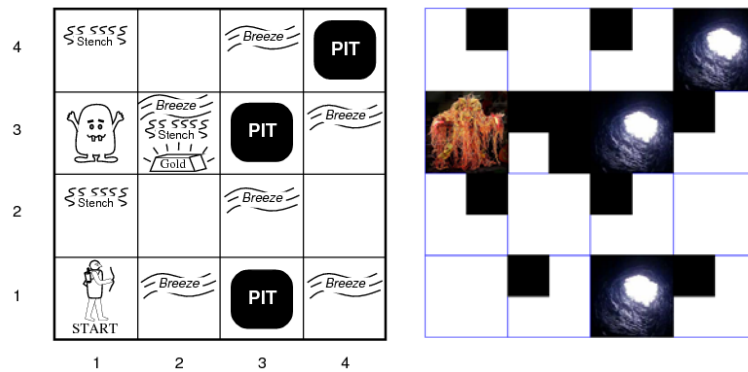


Figura 25 – “O Mundo de Wumpus” – O original e o modificado

A resolução do projecto estava dividida em duas fases. A primeira foi a construção do robot e a segunda a construção do programa para resolver o problema.

Na primeira fase, os alunos eram livre de construir o seu robot e os resultados foram em muitos casos, absolutamente originais.



Figura 26 – Exemplos de robots construídos pelos alunos

Em relação à segunda fase a experiência foi um sucesso e os resultados obtidos mostraram-se promissores. A seguir mostramos um excerto de código produzido por um grupo de alunos.

```
main(N,P):-inicializa(N,T), executa(N,P,T,T1,G), fim(N,T1,G).
inicializa(N,T) :-
    write('Bem vindo a Caverna do Wumpus...'), M is N*N,
    cria_mundo(M,T).
% Deus criou o mundo em 7 dias, nós apenas usamos duas linhas %(a
recursividade é divina);-)...
cria_mundo(0,[]) :- !.
cria_mundo(N,[0|T]) :- N1 is N-1, cria_mundo(N1,T).
executa(N,Pos,Ti,Tf,G):- ler_sensores(A,B,C,Pos),
    avalia_mundo(N,Pos,A,B,C,Ti,T1),
    mudar_lugar(4,Pos,T1,T2),
```

```

mostrar_caverna(N,T2),
explorar_mundo(N,Pos,T2,Tf,C,G).

% Avalia o Mundo usando a informação dos sensores
avalia_mundo(N,Pos,A,B,C,Ti,Tf) :-
    avalia_frente(N,Pos,A,B,C,Ti,T1),
    avalia_direita(N,Pos,A,B,C,T1,T2),
    avalia_esquerda(N,Pos,A,B,C,T2,T3),
    avalia_tras(N,Pos,A,B,C,T3,Tf).

avalia_frente(N,Pos,_,_,_,T,T):-      (Pos//N) == (N-1), !.
avalia_frente(N,Pos,A,B,C,Ti,Tf) :-
    Nxt is Pos+N, get_pos(Nxt,Ti,Val),
    verifica_perigo(Val,A,B,C,Ans),
    mudar_lugar(Ans,Nxt,Ti,Tf).

% idem direita, esquerda, tras
% explorar o mundo
explorar_mundo(_,_,T,T,1,1) :-
    !, nl, nl, write('Encontrei o tesouro! Ah! Ah!').
explorar_mundo(N,Pos,Ti,Tf,0,G) :-
    explorar_frente(N,Pos,Ti,T1,0,G1),
    explorar_direita(N,Pos,T1,T2,G1,G2),
    explorar_esquerda(N,Pos,T2,T3,G2,G3),
    explorar_tras(N,Pos,T3,Tf,G3,G).

explorar_frente(_,_,T,T,1,1) :- !.
explorar_frente(N,Pos,T,T,0,0) :-      (Pos//N) == (N-1), !.
explorar_frente(N,Pos,Ti,Tf,0,G) :-
    Nxt is Pos+N,
    get_pos(Nxt,Ti,Ans),
    Ans == 0, !,
    go_frente2,
    executar(N,Nxt,Ti,Tf,G),
    go_tras.

explorar_frente(_,_,T,T,G,G).
% idem direita, esquerda, tras
% Define o moviemnto do Robot:
go_frente2 :-      nl, nl,
    write('Avançar.'),
    acancar(X,Y),
    go_frente(X,Y,_).

avancar(1,70).
% idem go_tras2, virar_esquerda2,virar_direita2

```

Tabela 13 – Excerto de código “O Mundo de Wumpus”



### **7.2.2. 2006/2007 – “O Problema dos Canibais e Missionários”**

Para o ano lectivo 2006/2007, o projecto proposto aos estudantes foi uma adaptação do problema clássico dos canibais e missionários.

O projecto foi desenvolvido por grupos compostos por três alunos.

O problema define-se da seguinte forma:

Um grupo de missionários e canibais estão na margem de um rio e pretendem atravessar para o outro lado. O número de missionários e canibais são iguais e podem ser três, quatro ou cinco.

Eles têm um barco para fazer a travessia. O barco tem a capacidade para duas pessoas no caso de serem três missionários/canibais e três pessoas nos casos de quatro ou cinco missionários/canibais.

Se durante a travessia estiver mais canibais na margem dos missionários, os canibais comem-nos.

O objectivo dos grupos era enviar o robot para terminar a travessia. Para tal o robot tem de ter a mesma capacidade do barco e pelo menos uma pessoa tem de estar no barco em cada travessia.

Foi proposto que as margens do rio fossem representadas por linhas pretas, os canibais por bolas vermelhas e os missionários por bolas azuis.

A resolução deste problema estava dividida em três fases, nomeadamente a construção do robot, a preparação do meio ambiente e por fim a programação do robot. Em relação à primeira fase os alunos eram livres de escolher a forma do robot, para resolver o problema.



Figura 27 – Alguns robots propostos

No que diz respeito ao meio ambiente, este estava fortemente ligado à construção do robot. Contudo, os alunos poderiam fazer alterações ao meio ambiente inicialmente proposto desde que mantivessem o mesmo contexto.

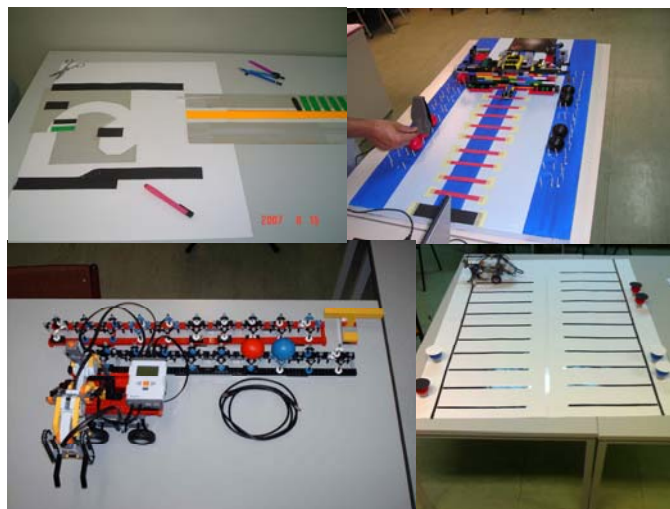


Figura 28 – Ambientes diferentes para resolver o problema

Em relação à programação do robot, os alunos usaram técnicas de Inteligência Artificial, no caso particular, pesquisas heurísticas. A seguir mostramos um excerto de código produzido por um grupo de alunos.

```

%%%-
%%-  contar bolas vermelhas com o sensor1 ate linha final
%%%-
removevermelhos1(X,[], []).
removevermelhos1(X, [X/L], L).
%%removevermelhos1(X, [Y/L], [Y/LR]):-X\==Y, removevermelhos1(X,L,LR).

```

```

conta_listavermelhos1([],T, T).
conta_listavermelhos1([X|L],R,S):-
X==vermelho,R1 is R +1 ,conta_listavermelhos1(L,R1,S).
conta_listavermelhos1([X|L],R,S):-
X\==vermelho,R1 is R+0, conta_listavermelhos1(L,R1,S).
andavermelhos1(Lista,V):- recuar_ate_sensor_luzvermelha31(50,50,Res),
luz1(X),X==vermelho,luz3(Y),Y==n1,recuar_ate_sensor_luz1(30,46,Res),
removevermelhos1(X,Lista,L),andavermelhos1(L,V),!.
andavermelhos1(L,V):-conta_listavermelhos1(L,0,D),V is 5-D.
%%%-----
%%%      contar bolas vermelhas com o sensor2 ate linha final
%%%-----
removevermelhos2(X,[], []).
removevermelhos2(X, [X|L], L).
%%removevermelhos2(X, [Y|L], [Y|LR]):-X\==Y, removevermelhos2(X,L,LR).
conta_listavermelhos2([],T, T).
conta_listavermelhos2([X|L],R,S):-
X==vermelho,R1 is R +1 ,conta_listavermelhos2(L,R1,S).
conta_listavermelhos2([X|L],R,S):-
X\==vermelho,R1 is R+0, conta_listavermelhos2(L,R1,S).
andavermelhos2(Lista,V):-
recuar_ate_sensor_luzvermelha32(50,50,Res),luz2(X),
X==vermelho,luz3(Y),Y==n1,recuar_ate_sensor_luz2(30,46,Res),
removevermelhos2(X,Lista,L),andavermelhos2(L,V),!.
andavermelhos2(L,V):-conta_listavermelhos2(L,0,D),V is 5-D.
%%%-----%%%
%%%      Algoritmo - Caminho mais curto      %%%
%%%-----%%%
read_data(File):-      open(File,read,S),
                        read(S,X),
                        assert(X),
                        write(X),
                        read_data_lp2(S,X),
                        close(S).
read_data_lp2(_S,end_of_file):-!.
read_data_lp2(S,_):-      read(S,Y),
                        assert(Y),
                        write(Y),
                        read_data_lp2(S,Y).
ha_caminho(A, B) :- ligacao(A, B, _), !.
ha_caminho(A, B) :- ligacao(A, X, _),

```

```

ha_caminho(X, B).
travessia(A, B, Visitados, [B|Visitados]) :- ligacao(A, B, _).
travessia(A, B, Visitados, Cam) :-
    ligacao(A, C, _),
    C \== B,
    \+ member(C, Visitados),
    travessia(C, B, [C|Visitados], Cam).
/* predicado que determina, aleatoriamente, um caminho entre A e B */
caminho(A, B, Cam) :- travessia(A, B, [A], Cam).
/* determinação de todos os caminhos entre dois nodos A e B */
caminhos(A, B, Lc) :- setof(Cam, caminho(A, B, Cam), Lc), !.
caminhos(_, _, []).
...
% pesquisa em largura
% dado uma lista de caminhos candidados
% 1. se a meta eh a cabeca da lista, o caminho eh a solucao
% 2. pega o primeiro caminho, gera todas as geracoes de filhos
% e inclui os novos caminhos no final dos abertos

solucao(Inicio, Fim, Solucao) :- buscal([ [Inicio] ], Fim, Solucao).
% caso 1.
buscal( [ [Meta|Solucao] | OutrosCaminhos], Meta, [Meta|Solucao]).
% caso 2.
buscal([ Caminho|RAbertos], Meta, Solucao) :-
    filhos( Caminho, Filhos),
    conc( RAbertos, Filhos, NovoAbertos),
    buscal( NovoAbertos, Meta, Solucao).
% gera os filhos de um caminho
filhos([n(AcaoN,Nodo) | Caminho], NovosCaminhos) :-
    findall([n(NomeAcao,Filho), n(AcaoN,Nodo) | Caminho],
    (acao(NomeAcao, Nodo, Filho), not(member(n(_,Filho),
    [n(_,Nodo)|Caminho]))),
    NovosCaminhos).
...

```

Tabela 14 - Excerto de código “Os Canibais e Missionários”

## 7.3. Conclusão

---

Embora não fosse nosso objectivo aplicar os robots na sala de aula e analisar dados resultantes dessa aplicação, estas experiências permitiram-nos ter uma noção da utilidade da aplicação do trabalho por nós apresentado.

Em relação à acção de formação (secção 7.1) permitiu-nos contextualizar e ambientar os professores e futuros professores na manipulação de uma nova tecnologia, neste caso o kit da Lego® Mindstorms™. Com esta acção conseguimos despertar o interesse dos presentes, motivando-os para a utilização destas tecnologias na sala de aula.

Devido à experiência dos presentes na acção podemos constatar que as propostas apresentadas por nós seriam uma mais valia, o que de alguma forma deu-nos algum alento e suporte para a maioria do trabalho desenvolvido.

Os problemas que foram resolvidos na terceira parte da acção foram apresentados com um formato aproximado aos apresentados no capítulo 6. Com isto pudemos testar a organização e validar algumas das opções tomadas.

Quanto à aplicação dos robots em conjunto com a plataforma RCX+PROLOG, tem se mostrado uma mais valia para a disciplina, tanto pela aplicabilidade prática na resolução dos projectos, como na motivação dos alunos (alguns resultados interessantes sobre esta questão podem ser consultados num artigo recentemente publicado [FF2007]).

Estas experiências permitiram-nos testar e refinar a plataforma RCX + PROLOG, pelo que neste momento temos uma plataforma estável para a versão RCX da Lego® Mindstorms™.

No capítulo seguinte apresentamos algumas conclusões e perspectivas de trabalhos futuros de todo o trabalho desenvolvido e apresentado neste documento.

---

## 8. Conclusão e Trabalhos Futuros

---

*“I hear and I forget. I see and I remember. I do and I understand.”*

**Confucius**

Este capítulo apresenta as conclusões gerais de todo o trabalho de investigação relativamente à concepção dos problemas e plataformas de programação para usar o robot como mediador entre o aluno e a programação. O capítulo identifica e descreve ainda alguns aspectos em aberto que merecem trabalho futuro.

### 8.1. Conclusão

---

O uso de robots no ensino tem vindo a afirmar-se como uma ferramenta muito útil, em especial nas áreas científicas e tecnológicas. Contudo é real a falta de estudos na preparação de plataformas de programação e de problemas para serem usados em contexto nas aulas. Identificamos, desde logo, os seguintes objectivos, que serviram de guia para o nosso trabalho de investigação:

- Criar problemas para serem resolvidos, usando os robots, nas disciplinas de programação do ensino secundário;
- Analisar, optimizar e desenvolver plataformas para serem aplicadas nas disciplinas onde os robots possam ser usados;

- Desenvolver uma plataforma para programar os robots nas aulas de Inteligência Artificial do ensino superior, usando a Linguagem Prolog;
- Criar materiais de suporte para alunos e professores das ferramentas/soluções desenvolvidas.

Em relação aos problemas apresentados no capítulo 6, e após análise cuidada aos programas das disciplinas de Informática do Ensino Secundário, a escolha recaiu na disciplina de Bases de Programação 10.º ano, pelo facto de ser uma disciplina base do ensino da programação.

Criamos uma série de problemas enquadrados nos conceitos da disciplina que permitirão uma aprendizagem evolutiva. Estes problemas são uma mais valia, pois os robots possibilitam resultados visíveis e concretos, motivando os alunos para a aprendizagem de conceitos mais avançados.

Como consequência da resposta ao objectivo anterior, surgiu-nos uma necessidade eminente: tínhamos os problemas e a disciplina onde aplicar esses problemas, faltava-nos a ferramenta que permitisse resolver esses problemas usando os robots e a linguagem de programação proposta pelo programa da disciplina.

Do estudo efectuado sobre as ferramentas, linguagens de programação e firmware's existentes no mercado, verificamos que poderíamos integrar diversas tecnologias para dar resposta a esta necessidade. Criamos então a plataforma RCX + Pascal, apresentada no capítulo 5, que permite a programação do robot usando a linguagem Pascal.

Ainda neste contexto das plataformas, e para dar resposta ao terceiro objectivo estabelecido, criamos a plataforma RCX + PROLOG. Esta plataforma permitiu, por um lado, a introdução dos robots nas aulas de Inteligência Artificial e, por outro, a possibilidade de programar esses robots usando a linguagem Prolog.

Esta plataforma já foi usada nas aulas de Inteligência Artificial (como descrito no capítulo 7) e a sua utilização em conjunto com os robots promoveu a discussão de ideias entre os alunos, bem como a colaboração e cooperação para chegar a uma solução dos problemas propostos.

O facto de já termos aplicado esta plataforma em dois anos lectivos permitiu-nos corrigir e refinar questões de implementação, evoluindo a plataforma para uma versão estável e bem documentada.

A grande vantagem destas plataformas advém do facto de seguirem a mesma filosofia de programação existente na plataforma nativa RIS, ou seja, em todas elas existem o conceito dos “blocos” de programação, o que cria um padrão na forma de programar o robot.

Em relação ao último objectivo, que naturalmente surge como consequência dos anteriores, criamos uma série de materiais de suporte à utilização, dos alunos e professores, das ferramentas desenvolvidas. Esses materiais foram:

- Guia de Instalação e Utilização da plataforma RCX+PROLOG (ver anexo B ou disponível em <http://dme.uma.pt/droide/rcxprolog>);
- Guia de Instalação e Utilização da plataforma RCX+Pascal (disponível em <http://dme.uma.pt/droide/rcxpascal>);
- Manual da API de Pascal (ver anexo A ou disponível em <http://dme.uma.pt/droide/rcxpascal>);
- CD de instalação da plataforma RCX+PROLOG (disponível em <http://dme.uma.pt/droide/rcxprolog>);
- CD de instalação da plataforma RCX+Pascal disponível em <http://dme.uma.pt/droide/rcxpascal>).

## **8.2. Trabalhos Futuros**

---

A realização de qualquer trabalho de investigação encontra-se necessariamente incompleta e suscita linhas em aberto que poderão ser alvo de trabalho futuro. Numa perspectiva objectiva e pragmática identificam-se os seguintes aspectos principais:

- Aplicação dos problemas apresentados em situações reais de sala de aula e análise dos resultados;
- Migração das plataformas apresentadas, para a nova versão dos robots da Lego®.

Segue-se uma breve descrição dos possíveis trabalhos futuros acima enumerados.



### **Aplicação dos problemas apresentados em situações reais de sala de aula e análise dos resultados**

Como já referido, não era nosso objectivo aplicar os problemas e analisar os resultados que surgissem dessa aplicação. Contudo, sugerimos vivamente este trabalho, pois acreditamos que será uma mais valia para o ensino da programação em níveis iniciais, como o caso da disciplina de Bases de Programação de 10.º ano.

Como resultado da acção de formação desenvolvida, apresentada no capítulo 7, existe já professores interessados na sua utilização, pelo que serão desenvolvidos esforços neste sentido.

### **Migração das plataformas apresentadas, para a nova versão dos robots da Lego®**

Dada a iminente descontinuação já anunciada pela Lego® da produção do modelo RCX em detrimento da versão denominada por NXT [NXT], será necessário como trabalho futuro, de forma a dar continuidade ao trabalho e estudos em curso, a migração das plataformas apresentadas para este novo modelo.

---

# Referências

---

- [AFF2007] Alcione, E., Fermé, E., Fernandes, E., *DROIDE VIRTUAL: UTILIZAÇÃO DE ROBOTS NA APRENDIZAGEM COLABORATIVA DA PROGRAMAÇÃO ATRAVÉS DA WEB*, 2007.
- [ARJ] API RCX Java : <http://www.escape.com/~dario/java/rcx/>.
- [B2002] David Barnes, *Teaching Introductory Java through LEGO MINDSTORMS Models*, SIGCSE, 2002.
- [B2003] Dave Baum, *NQC programmer's Guide*, 2003.
- [BCC] Bricx Command Center: <http://bricxcc.sourceforge.net/>.
- [BOS] BrickOS: <http://brickos.sourceforge.net/>.
- [C2002] Chella, M. T., *Ambiente de Robótica para Aplicações Educacionais com SuperLogo*. Universidade Estadual de Campinas – UNICAMP. Faculdade de Engenharia Elétrica e da Computação – FEEC. Dissertação de mestrado, 2002.
- [CCM2006] Célia R., Clara C., Manuel C., *RobotCarochinha: Um Estudo sobre a Robótica Educativa no Ensino Básico*, 2006.
- [D2005] *DROIDE: Os Robots como Elementos Mediadores entre o Aluno e a Matemática/Informática. Projecto do Departamento de Matemática e Engenharias da Universidade da Madeira*. 2005 Madeira.
- [DH2000] Druin, A., Hendler, J., *Robots for kids: Exploring new technologies for learning*, San Diego, CA: Academy Press, 2000.
- [DVL2004] Douglas M., Viviane A., Luiz G., *Em Evidência o Potencial e Limitações dos Compiladores NQC e BRICKOS e seus Respectivos Sistemas Operacionais*, Revista de Informática Teórica e Aplicada, 2004.

- [FF2002] Ferrari, M., G. Ferrari, and R. Hempel, *Building Robots With LEGO Mindstorms*. 2002: Syngress Publishing, Inc.
- [FF2007] Fermé, F., Fernandes, F., *Crossing the river with robots: Changing the way of working in an AI subject*, Proceedings of the Artificial Intelligence in Education Workshop, Cambridge UK, 2007.
- [FFA2007] Fermé, E., Fernandes, E., Alcione, E., *UTILIZAÇÃO DE ROBOTS NO ENSINO DE PROGRAMAÇÃO: O PROJECTO DROIDE*, 2007.
- [FFO2006] Fernandes, E., Fermé, E., Oliveira, R., *Using Robots to Learn Functions in Math Class*, Proceedings of the ICMI 17 Study Conference: Technology Revisited. Hanoi University of Technology, 2006.
- [FFO2007] Fernandes, E., Fermé, E., Oliveira, R., *Viajando com Robots na Aula de Matemática*, Proceedings of the V Conferência Internacional de Tecnologias de Informação e Comunicação na Educação, Minho, Portugal, Challenges 2007.
- [FG1997] Franklin, S., Graesser, A., *Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents*. 1997.
- [FG2007] Fermé, F., Gaspar, L., *RCX+PROLOG: A platform to use Lego Mindstorms(tm) Robots in Artificial Intelligence courses*, Proceedings of the Artificial Intelligence in Education Workshop, Cambridge UK, 2007.
- [FGH+2002] Ferrari, G., Gombos, A., Hilmer, S., Stuber, J., Porter, M., Waldinguer, J., e Laverde, D., *Programming Lego Mindstorms With Java*. 2002: Syngress Publishing, Inc.
- [FME2001] Fagin, B., Merkle, L. D., Eggers, T. W., *Teaching computer science with robotics using Ada/Mindstorms 2.0*. In Proceedings of the 2001 Annual ACM SIGAda International Conference on Ada. New York: ACM Press, (pp. 73-78).
- [GSP+2002] Guerra, L., Simões F., Pinto M., Silva, M., Rodrigues, L., *Programa de Bases de Programação 10º, 11º e 12º anos - Curso Tecnológico de Informática*. 2002.
- [HW2005] Han, K. S., Wook, J. J., *Learning C Language Using Robots*, ICCAS, 2005.

- [J2003] João, S.M., *Programa de Tecnologias de Informação e Comunicação 9.º e 10.º Anos*. 2003.
- [JCA] Java Communication API: <http://java.sun.com/products/javacomm/>
- [JM2003] Joan M., Mike C., *Designing, developing and Implementing a Course on LEGO Robotics for Technology Teacher Education*, JI. Of Technology and Teacher Education, 2003.
- [JPD2004] João, S.M., Pinto, M.L.S., Dias, P.M., *Programa de Aplicações Informáticas A - Curso Tecnológico de Informática 10º Ano*. 2004.
- [K1979] Kowalsky, R., *Logic for Problems*, Elsevier North-Holland, 1979.
- [K2005] Amruth N. Kumar, *Three Years of Using Robots in the Artificial Intelligence Course – Lessons Learned*, Journal of Educational Resources in Computing, 2005.
- [LM] Lego® Mindstorms™:  
<http://www.lego.com/eng/education/mindstorms/default.asp>
- [LOS] LejOS: [http://lejos.sourceforge.net/p\\_technologies/rcx/lejos.php](http://lejos.sourceforge.net/p_technologies/rcx/lejos.php).
- [LVIEW] LabVIEW: <http://www.ni.com/labview/>
- [LP2000] Levesque, H.J., M. Pagnucco, *Legolog: Inexpensive Experiments in Cognitive Robotics*, 2000.
- [MSE+2002] Martha C., Scott M., Elissa M., Merredith P., Chris R., *ROBOLAB Reference Guide Version 1.2*, LEGO Educational Division, 2002.
- [NXT] Lego® Mindstorms™ NXT: <http://mindstorms.lego.com/Overview/>
- [OHM1999] Orazio M., Henrik L., Maurizio C., *Robotics as na Educational Tool, JI. Of Interactive Learning Recherche*, 1999.
- [PDJ2005] Pinto, M.L.S., Dias, P.M., João, S.M., *Programa de Aplicações Informáticas A - Curso Tecnológico de Informática 11º Ano*. 2005.
- [PS2004] Parsons, S., Sklar, E., *Teaching AI using LEGO Mindstorms*, Proceedings of the AAAI Spring Symposium on Accessible Hands-on Artificial Intelligence and Robotics Education, Stanford, 2004.

- [PSD+2006] Pinto, M., Silva, M., Dias, P., Moreira, P. João, S., *Programa de Técnicas de Gestão de Bases de Dados 12.º ano - Curso Tecnológico de Informática*. 2006.
- [RIS] Robotics Ivention System™:  
<http://mindstorms.lego.com/eng/products/ris/index.asp>
- [RP2004] Rogers, C., Portsmore, M. (2004). *Bringing Engineering to Elementary School. Journal of STEM Education*, 5(3,4).
- [RJA] RCX JAVA API: <http://www.escape.com/~dario/java/rcx/>.
- [RN2003] Russell, S.J. and P. Norvig, *Artificial Intelligence A Modern Approach*. Second Edition. Prentice Hall, Englewood Cliffs, New Jersey. 2003.
- [SCF2001] Soares, L.F., Cruz, J.A., Fernandes, V.M., *Programa de tecnologias Informáticas 10º - 11º - 12º Anos - Curso Tecnológico de Informática*. 2001.
- [W2002] Wooldridge, M. *An Introduction to Multiagent Systems*. John Wiley, 2002.
- [P1980] Seymour Papert, *Mindstorms: Children, computers and powerful ideas*, New York: Basic Books, 1980.
- [Z2004] Silvana R. Zilli, *A ROBÓTICA EDUCACIONAL NO ENSINO FUNDAMENTAL: PRESPECTIVAS E PRÁTICA*, Universidade Federal de Santa Catarina – Dissertação de Mestrado, 2004.
- [W2001] Wong, Ka-Wing, *Teaching Programming with Lego RCX Robots*, Proceedings of the 18th Information Systems Education Conference (ISECON 2001), 2001

---

# **Anexos**

---

---

## **Anexo A - Manual da API de Pascal**

---

## Introdução

Este documento apresenta um manual da “API Pascal” para os robots da Lego® Mindstorms™. Este manual foi criado para que os alunos possam desenvolver programas para os robots, nas suas aulas, usando a linguagem Pascal.

São apresentados ao longo do documento exemplos, que ajudam a compreensão do funcionamento das várias funções da API, permitindo uma melhor visualização do uso dessas funções.

## 1. Temporizadores

Uma das primeiras funções a ter em conta na programação do seu RCX é a função `sleep(«segundos»);`. Esta função é responsável por manter o comportamento das últimas instruções durante um determinado tempo em segundos. Por exemplo, é muito importante para indicar a duração que um motor deve funcionar.

Se pretender indicar tempos mais curtos do que um segundo, ou tempos como por exemplo 1,5 segundos, poderá usar a função `msleep(«msegundos»);`, que determina tempos na ordem dos milésimos de segundo. Por exemplo, a instrução `msleep(1500);` representa 1,5 segundos e a instrução `msleep(1);` representa 1 milésimo de segundo.

As duas funções anteriores estão definidas na biblioteca `unistd`, pelo que deve incluí-la no seu programa para que possam ser executadas.

## 2. Motores

Podemos ligar até 3 motores com comportamento independente ao nosso RXC, nas portas A, B e C, respectivamente. Para transmitir comportamento aos motores, tem de ser incluída a biblioteca `dmotor`. Nesta biblioteca estão implementadas as seguintes funções:

**`motor_a_dir(«parâmetro»);`** - Função responsável pelo comportamento do motor A.

**Nota:** Para aplicar um comportamento aos motores B e C, deverá usar as funções **`motor_b_dir(«parâmetro»);`** e **`motor_c_dir(«parâmetro»);`** respectivamente.

«parâmetro» poderá ser uma das seguintes constantes:

- **`mdFwd`** - Permite o motor avançar;
- **`mdRev`** - Permite ao motor recuar;
- **`mdOff`** - Desliga a corrente do motor, ou seja para o motor.
- **`mdBrake`** - Trava o motor num determinado momento.

**`motor_a_speed(«valor»);`** - Estabelece a velocidade com que o motor A vai avançar ou recuar. O valor dessa velocidade varia no intervalo de 0 a 250.

Ainda em relação à velocidade estão definidas duas constantes:

- **`MIN_SPEED`** → Representa o Valor 0;



- MAX\_SPEED → Representa o Valor 255;

**Nota:** Para aplicar um comportamento aos motores B e C, deverá usar as funções ***motor\_b\_speed(«valor»)***; e ***motor\_c\_speed(«valor»)***; respectivamente.

**Exemplo:** O programa seguinte permite avançar o Motor A e C em simultâneo durante dois segundos.

```
program avanca1;

uses
  unistd, dmotor;

begin
  motor_a_dir(mdFwd);
  motor_c_dir(mdFwd);
  motor_a_speed(255);
  motor_c_speed(255);
  sleep(2);
  motor_a_dir(mdOff);
  motor_c_dir(mdOff);

end.
```

### 3. Sensores

Existem três portas de ligação para os sensores, numeradas no RCX por 1, 2 e 3. A cada uma destas portas existe uma constante associada, nomeadamente ***SENSOR\_1***, ***SENSOR\_2***, ***SENSOR\_3***. Estas constantes podem ser usadas como argumentos em qualquer uma das funções que tratem dos sensores.

A **API** suporta três tipos de sensores, ou seja, permite nos seus programas usar três tipos de sensores:

1. Sensor de toque;
2. Sensor de luz;
3. Sensor de rotação;

#### 3.1. Sensor de toque

O Sensor de toque tem dois estados, Activo ou Inactivo. A função responsável pela programação do sensor de toque é a função ***TOUCH(«porta\_de\_ligação»)***; do tipo Booleano, ou seja retorna Verdadeiro (***TRUE***) caso o sensor seja activado, Falso (***FALSE***) caso contrário. Esta função pertence à biblioteca ***dsensor***.

Caso o sensor esta ligado à porta 1 do seu RCX, a função seria aplicada nessa porta, ou seja, ***TOUCH(SENSOR\_1)***; Para o caso de o sensor estar ligado noutra porta, a função seria aplicada à constante correspondente como mostra o exemplo seguinte.

**Exemplo:** Se o sensor de toque ligado à porta 3 estiver activado, então o robot avança um segundo.

```

program se_toque_3;

uses
  unistd, dsensor, dmotor;

begin

  if( TOUCH( SENSOR_3 )=TRUE) then

    begin
      motor_a_dir(mdFwd);
      motor_c_dir(mdFwd);
      motor_a_speed(255);
      motor_c_speed(255);
      sleep(1);
      motor_a_dir(mdOff);
      motor_c_dir(mdOff);

    end;
  end.

```

### 3.2. Sensor de luz

O sensor de luz mede a quantidade de luz num determinado ponto. O sensor de luz emite também uma luz, que possibilita apontar o sensor numa determinada direcção e medir as diferenças da intensidade de luz num determinado objecto. Tal como na secção anterior as funções para programar este sensor estão na biblioteca **dsensor**.

Para activar o sensor deve usar a seguinte função: **ds\_active(@«Porta\_de\_Ligação»);**

O parâmetro «**Porta\_de\_Ligação**» representa SENSOR\_1 ou SENSOR\_2 ou SENSOR\_3, dependendo da porta onde esteja ligado.

A função **LIGHT(«Porta\_de\_Ligação»);** retorna o valor correspondente a intensidade da luz num intervalo de [0-100].

O exemplo a seguir mostra a comparação do sensor ligado na porta 2 com um determinado valor. Se a intensidade da luz corresponder ao número 39 então o robot avança.

```

program valorluz;

uses
  unistd, dsensor, dmotor;

begin
  ds_active(@SENSOR_2);

  if(LIGHT(SENSOR_2)=39) then
    begin
      motor_a_dir(mdFwd);
      motor_c_dir(mdFwd);
      motor_a_speed(255);
      motor_c_speed(255);
      sleep(2);
      motor_a_dir(mdOff);
      motor_c_dir(mdOff);

    end;
  end.

```

Poderá ainda usando a função **ds\_passive(@«Porta\_de\_Ligação»);** para desactivar o sensor de luz, ou seja, depois da execução desta instrução o sensor de luz deixa de emitir luz e passa a efectuar apenas leituras da luz ambiente.

### 3.3. Sensor de rotação

O Sensor de rotação mede, como o nome indica, rotação. Este sensor por cada volta passa por estados, isto quer dizer que em cada volta ele conta 16 unidades. Para além disto, podemos ainda através do sensor de rotação determinar o sentido da sua rotação, ou seja, num sentido incrementa as unidades por volta e para o outro decrementa esse valor. Esta contagem é relativa ao ponto de partida do sensor. A contagem está limitada ao intervalo de valores de -32767 até 32767. As funções seguintes estão definidas na biblioteca **dsensor**.

Tal como o sensor de luz, este precisa de ser activado. Para tal deve de usar a função **ds\_active(@«Porta\_de\_Ligação»);** O parâmetro «Porta\_de\_Ligação» representa SENSOR\_1 ou SENSOR\_2 ou SENSOR\_3, dependendo da porta onde esteja ligado.

A função **ds\_rotation\_set(@«Porta\_de\_Ligação»; «valor»);** atribui o valor inicial da contagem do sensor de rotação. Por este facto é que a contagem, como referido anteriormente, é relativa ao ponto de partida do sensor. Normalmente este valor é 0.

Depois de activado o sensor é preciso ligá-lo. A função **ds\_rotation\_on(@«Porta\_de\_Ligação»);** liga o sensor de rotação, iniciando a contagem. Por sua vez a função **ds\_rotation\_off(@«Porta\_de\_Ligação»);** desliga o sensor de rotação.

Contudo, para que possamos obter os valores do sensor de rotação precisamos recorrer às seguintes funções:

**ROTATION\_1;** - Retorna os valores do sensor de rotação quando ligado à porta 1.

**ROTATION\_2;** - Retorna os valores do sensor de rotação quando ligado à porta 2.

**ROTATION\_3;** - Retorna os valores do sensor de rotação quando ligado à porta 3.

O exemplo a seguir mostra no LCD do RCX os valores do sensor de rotação com um espaçamento temporal de meio segundo.

```
program valorrot;  
  
uses  
    unistd, dsensor, romlcd;  
  
var v:integer;  
  
begin  
    ds_active(@SENSOR_2);  
  
    //Inicializa o sensor a 0.  
    ds_rotation_set(@SENSOR_2, 0);  
    ds_rotation_on(@SENSOR_2);  
  
    while(TRUE) do  
        begin  
  
            //atribui o valor à variável v  
            v:=ROTATION_2;  
            lcd_int(v);  
            msleep(500);  
        end;  
  
    end.
```

## 4. Som

Para que o robot transmita sons tem de incluir a biblioteca *dsound*. Nesta biblioteca encontra a função **dsound\_system(«valor»)**; Esta função, com o valor zero, liberta um som conhecido por beep. Poderá ainda usar a constante **DSOUND\_BEEP** como valor da função. Exemplo **dsound\_system(DSOUND\_BEEP)**;

## 5. LCD

Para mostrar valores e/ou informação no LCD do RCX, tem de incluir as bibliotecas **romlcd** e **conio**.

Da biblioteca **romlcd** pode usar as seguintes funções:

- **lcd\_int(«valor»)**; - mostra no LCD o «valor» no formato de número inteiro.
- **lcd\_unsigned(«valor»)**; - mostra no LCD o «valor» no formato de número real, com duas casas decimais.

Por outro lado da biblioteca **conio** podem ser usadas as seguintes funções:

- **cputs('string')** – Mostra no LCD uma string;

O exemplo a seguir mostra no LCD a frase “Ola Mundo”. As duas palavras aparecem separadas por um segundo no LCD.

```
program olamundo;

uses
    conio, unistd;

begin
    cputs('Ola');
    sleep(1);
    cputs('Mundo');
    sleep(1);
end.
```

- **cputc('caracter',«valor»)** – Mostra um caracter, na posição n do LCD. Este valor varia no intervalo [1-4], onde 1 corresponde à posição mais à direita do LCD e o valor 4 à posição mais à esquerda.

O programa seguinte imprime no LCD a palavra “LUIS”.

```
program imprimec;

uses
    conio, unistd;

begin
    cputc('L',4);
    cputc('U',3);
    cputc('I',2);
    cputc('S',1);
end.
```

- **cls** – Limpa o LCD

Ainda na biblioteca **lcd**, poderão ser usadas as seguintes funções:

- **lcd\_power\_on**: Liga o LCD
- **lcd\_power\_off**: Desliga o LCD

## 6. Mensagens

Um robot pode comunicar com outro através da porta de infravermelhos. O RCX pode reagir a mensagens, pode enviar mensagens a outro RCX e pode receber mensagens do computador. As funções responsáveis pela comunicação entre robots estão definidas na biblioteca *lnp*.

A função ***send\_msg***(«*mensagem*»); envia uma mensagem do tipo numérico no intervalo [0-255].

A função ***get\_msg***; recebe uma mensagem através da porta de infravermelhos que pode ser atribuída a uma variável do tipo inteiro, podendo assim ser usada ao longo do programa.

Por fim, a função ***clear\_msg***; apaga eventuais mensagens que estejam em memória.

O exemplo a seguir mostra a comunicação entre dois robots.

```
program enviormsg;  
  
uses lnp;  
  
begin  
    send_msg(1);  
  
end.
```

Programa responsável pelo envio de uma mensagem.

```
program recebemsg;  
  
uses unistd, dmotor, lnp;  
var a:integer;  
  
begin  
    clear_msg;  
    a:=get_msg;  
  
    if(a=1) then  
        begin  
            motor_a_dir(mdFwd);  
            motor_c_dir(mdFwd);  
            motor_a_speed(255);  
            motor_c_speed(255);  
            sleep(2);  
            motor_a_dir(mdOff);  
            motor_c_dir(mdOff);  
        end;  
    end;
```

Programa responsável pela recepção da mensagem e reacção de acordo com a mensagem

## 7. Funções Genéricas do RCX

### 7.1. Baterias

A função ***get\_battery\_mv*** retorna o valor da bateria em “milivolts”, ou seja, um valor inteiro que representa o valor da voltagem multiplicado por mil. Esta função está implementada na biblioteca ***battery***.

O exemplo a seguir mostra no LCD o valor da energia das baterias.

```
program Batt;  
  
uses conio, unistd, romlcd,  
battery;  
  
begin  
    cputs('BATR');  
    sleep(1);  
    lcd_int(get_battery_mv);  
  
end.
```

## Bibliografia

1. Ferrari, M., G. Ferrari, and R. Hempel, *Building Robots With LEGO Mindstorms*. 2002: Syngress Publishing, Inc.

---

## **Anexo B- Guia de Instalação e Utilização da Plataforma RCX+PROLOG**

---



## RCX + PROLOG: Guia de Instalação

Para instalar e configurar o pacote de software que vai permitir programar os robots utilizando a linguagem de programação Prolog, deverá seguir os passos a seguir discriminados:

### Passo 1: Instalação do RIS 2.0

Caso seja a primeira vez que vai utilizar o kit LEGO® MINDSTORMS™ instale o software Robotics Invention System™ 2.0. Este software é disponibilizado juntamente com o kit. Se já tiver este software instalado passe então para o **Passo 3**.



Fig.1 - Imagem do CD de Instalação do software Robotics Invention System™ 2.0

### Passo 2: Actualização do Firmware

Actualize o Firmware do robot, usando o software instalado no passo anterior. Escolha a opção Settings (assinalado a vermelho na Fig. 2) e no ecrã seguinte escolha Download Firmware (Fig. 3).



Fig. 2 – Ecrã Software RIS 2.0

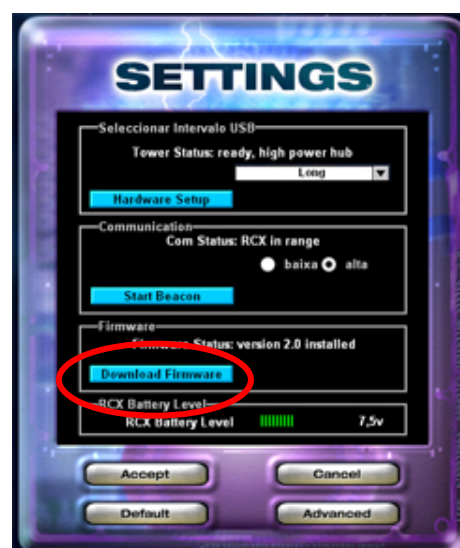


Fig. 3 – Definições

### Passo 3: Instalação JAVA 2 SDK

Instale o JAVA 2 SDK. Este software é necessário para poder compilar o ficheiro **Control.java**. Este ficheiro é responsável pela comunicação entre o Eclipse Prolog e o RCX.

Execute o ficheiro "**j2sdk-1\_4\_0\_01-windows-i586.exe**" para iniciar a instalação.

**Nota:** Encontra esta e outras versões em <http://java.sun.com/products/>

### Passo 4: Instalação do Tcl

Esta ferramenta possibilita uma plataforma para a integração de várias aplicações e neste caso concreto serve para instalar o Eclipse Prolog.

Execute o ficheiro "**tcl831.exe**" e instale este software na directoria de programas do seu computador.

**Nota:** Pode fazer o download em <http://www.tcl.tk/software/tcltk/>

### Passo 5: Instalação do Eclipse Prolog

Descompacte o ficheiro **eclipse.zip** para a raiz do seu disco (C:\).

Execute o ficheiro **ecl\_inst.tcl** para instalar o ECLiPSe no seu computador. Deverá indicar (Fig. 3) a directoria onde será instalado o eclipse, bem como a directoria de trabalho. Recomenda-se que a directoria de trabalho seja a mesma onde está localizado o ECLiPSe.

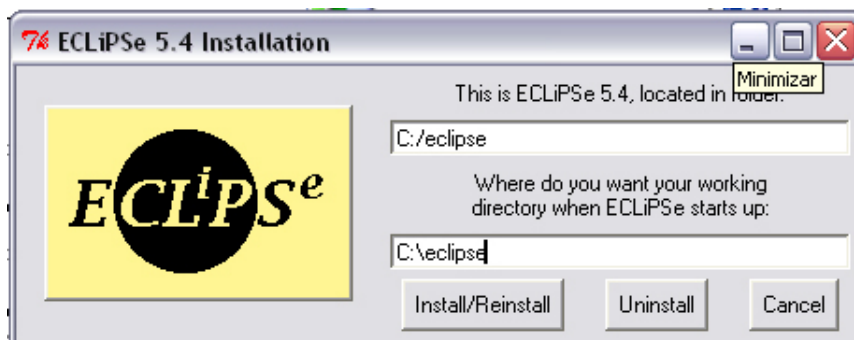


Fig. 3 – Instalação do Eclipse Prolog

**Nota:** Poderá encontrar mais informação e/ou versões deste software em:

<http://www.cs.hut.fi/Studies/T-93.540/eclipse/windowdownload.html>

### Passo 6: Instalar os pacotes de Comunicação

Instale na raiz do seu disco (C:\) a API "Java Communications Extension API" (ficheiro: **commapi.exe**) e a API "RCX Java API" (ficheiro: **rcxjava.exe**).

O RCX Java API é um pacote que providencia classes e interfaces, que suportam a comunicação com o RCX. Esta comunicação é suportada pela API de comunicação Java Communication API.

**Nota:** Poderá encontrar mais informação em:

<http://java.sun.com/products/javacomm/>

<http://www.escape.com/~dario/java/rcx/>

## Passo 7: Instalação do Compilador NQC

NQC "Not Quite C" é uma linguagem simples, com uma sintaxe parecida à Linguagem C, que é usada para programar o RCX.

Instale o compilador desta linguagem (ficheiro: **nqc.exe**), na raiz do seu disco (C:\), para poder fazer o download do programa para o seu RCX.

**Nota:** Poderá encontrar mais informação e/ou versões deste software em:

<http://bricxcc.sourceforge.net/nqc/>

## Passo 8: Configuração das variáveis de ambiente

Nas propriedades do sistema, no separador Avançadas, mude as seguintes variáveis de ambiente (Fig. 5):

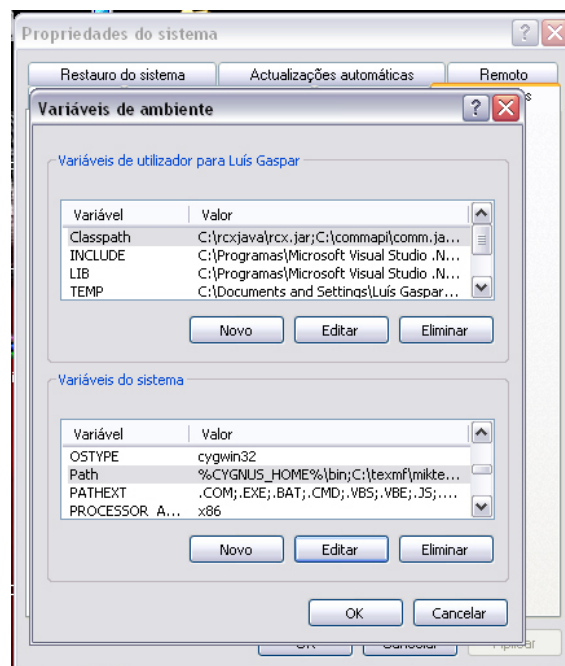


Fig. 5 – Variáveis de Ambiente

- **Path**

Tem que incluir: \$JAVADIR\bin; \$NQC\_DIR

Considerando que usou as directorias sugeridas ao longo dos vários passos, temos:

- \$JAVADIR = C:\j2sdk1.4.0\_01

- \$NQCDIR = C:\nqc

## • Classpath

Tem que incluir: \$RCXJAVAHOME\rcx.jar; \$COMMAPIHOME\comm.jar; \$ECLIPSEDIR\lib\eclipse.jar; \$WORKHOME

Considerando que usou as directorias sugeridas ao longo dos vários passos, temos:

- \$RCXJAVAHOME = C:\rcxjava
- \$COMMAPIHOME = C:\commapi
- \$ECLIPSEDIR = C:\eclipse
- \$WORKHOME = C:\eclipse

## Passo 9: Preparar o Ambiente de Trabalho

### 1- Copiar os ficheiros de Trabalho

Instale o ficheiro "**rcxprolog.exe**" para a sua directoria de trabalho, ou seja C:\eclipse.

Esta instalação copiará os seguintes ficheiros para a directoria de trabalho:

- Control.java
- controliia.nqh
- iia.nqc
- win32usb.dll
- win32com.dll
- main.pl

### 2- Compilar o ficheiro Control.java

**2.1-** Edite o ficheiro **Control.java** e na linha de código

```
"eclipse = new RemoteEclipse(InetAddress.getByName("computador"), 1567);"
```

mude "**computador**" para o nome do seu computador.

**2.2-** Compile o ficheiro, afim de criar a classe responsável pela comunicação entre o seu o ECLiPSe e o RCX. Numa janela de linha de comandos os seguintes passos:

- Colocar-se na sua pasta de trabalho (comando: **cd\eclipse**);
- Executar o comando **javac Control.java**, para compilar o ficheiro Control.java

### 3- Prepara o Robot para receber instruções

**3.1-** Conecte a torre USB do Kit ao computador e coloque a torre e o RCX de forma que o RCX consiga comunicar com a torre e vice-versa.

**3.2-** Faça o Download do programa **iia.nqc** para o RCX. Numa janela de linha de comandos execute os seguintes passos:

- Colocar-se na sua pasta de trabalho (comando: **cdleclipse**);
- Executar o seguinte comando: **nqc -Subb -d iia.nqc** e carregue em **RUN** no RCX

# RCX + PROLOG: Guia de Utilização

Para correr o seu programa, deverá seguir os seguintes passos:

## 1- Executar o programa escrito em Prolog

### 1.1- Edite o ficheiro main.pl e na linha de código

*"maquina(**computador**/1567)."*

mude "**computador**" para o nome do seu computador.

### 1.2- Inicie o Eclipse (Menu Iniciar -> Programas -> Eclipse 5.4 -> DosEclipse) e:

- Compile o ficheiro main.pl (comando: **compile('main.pl').** )
- Corra o predicado principal (comando: **main.** )

## 2- Executar a classe Control

Numa janela de linha de comandos execute os seguintes passos:

- Colocar-se na sua pasta de trabalho (comando: **cd\ eclipse** );
- Executar o seguinte comando: **java -Declipse.directory="C:\eclipse" Control**

Dará assim início a execução do seu programa.

---

## **Anexo C – Protocolo de Comunicação da Plataforma RCX + PROLOG**

---

<b>Cód.</b>	<b>Acção</b>	<b>Parâmetro 1</b>	<b>Parâmetro 2</b>
1	Avançar	Seg. [0-254]	Cent.[0-99]
2	Avançar Apenas Motor A	Seg. [0-254]	Cent.[0-99]
3	Avançar Apenas Motor B	Seg. [0-254]	Cent.[0-99]
4	Avançar Apenas Motor C	Seg. [0-254]	Cent.[0-99]
5	Avançar até Toque Sensor 1	-	-
6	Avançar até Toque Sensor 3	-	-
7	Avançar até Toque Sensor 1, 3	-	-
8	Avançar até Sensor de Luz	N.º Min [1-100]	N.º Máx.[1-100]
9	Avançar até Sensor de Luz, sensor de toque 1,3	N.º Min [1-100]	N.º Máx.[1-100]
10	Recuar	Seg. [0-254]	Cent.[0-99]
11	Recuar Apenas Motor A	Seg. [0-254]	Cent.[0-99]
12	Recuar Apenas Motor B	Seg. [0-254]	Cent.[0-99]
13	Recuar Apenas Motor C	Seg. [0-254]	Cent.[0-99]
14	Recuar até Toque Sensor 1	-	-
15	Recuar até Toque Sensor 3	-	-
16	Recuar até Toque Sensor 1, 3	-	-
17	Recuar até Sensor de Luz	N.º Min [1-100]	N.º Máx.[1-100]
18	Recuar até Sensor de Luz, sensor de toque 1,3	N.º Min [1-100]	N.º Máx.[1-100]
19	Pedir Valor do Sensor de Luz	-	-
20	Virar à Direita	Seg. [0-254]	Cent.[0-99]
21	Virar à Direita até Toque Sensor 1	-	-
22	Virar à Direita até Toque Sensor 3	-	-
23	Virar à Direita até Toque Sensor 1, 3	-	-
24	Virar à Direita até Sensor de Luz	N.º Min [1-100]	N.º Máx.[1-100]
25	Virar à Direita até Sensor de Luz, sensor de toque 1,3	N.º Min [1-100]	N.º Máx.[1-100]
26	Girar à Direita	Seg. [0-254]	Cent.[0-99]
27	Girar à Direita até Toque Sensor 1	-	-
28	Girar à Direita até Toque Sensor 3	-	-
29	Girar à Direita até Toque Sensor 1, 3	-	-
30	Girar à Direita até Sensor de Luz	N.º Min [1-100]	N.º Máx.[1-100]
31	Girar à Direita até Sensor de Luz, sensor de toque 1,3	N.º Min [1-100]	N.º Máx.[1-100]
32	Virar à Esquerda	Seg. [0-254]	Cent.[0-99]
33	Virar à Esquerda até Toque Sensor 1	-	-
34	Virar à Esquerda até Toque Sensor 3	-	-
35	Virar à Esquerda até Toque Sensor 1, 3	-	-
36	Virar à Esquerda até Sensor de Luz	N.º Min [1-100]	N.º Máx.[1-100]
37	Virar à Esquerda até Sensor de Luz, sensor de toque 1,3	N.º Min [1-100]	N.º Máx.[1-100]
38	Girar à Esquerda	Seg. [0-254]	Cent.[0-99]
39	Girar à Esquerda até Toque Sensor 1	-	-
40	Girar à Esquerda até Toque Sensor 3	-	-
41	Girar à Esquerda até Toque Sensor 1, 3	-	-
42	Girar à Esquerda até Sensor de Luz	N.º Min [1-100]	N.º Máx.[1-100]
43	Girar à Esquerda até Sensor de Luz, sensor de toque 1,3	N.º Min [1-100]	N.º Máx.[1-100]
44	Tocar um Beep	toque[1-6]	-
45	Tocar um Som	Freq.{5,10,15,20}	Seg. [1-5]
46	Desligar Motor A	-	-
47	Desligar Motor B	-	-
48	Desligar Motor C	-	-
49	Desligar Motor A,C	-	-



---

## **Anexo D – Problema usado na Acção de Formação.**

---



## Estruturas de Repetição

- 1- Usando uma estrutura de repetição, escreva um programa para que o robot seja capaz de avançar e recuar 5 vezes, finalizando com um som.
- 2- Usando uma estrutura de repetição, escreva um programa para que o robot avance até encontrar uma faixa de cor preta.

### Notas sobre a Actividade

<b>Enquadramento Curricular</b>	Esta actividade é aplicada na Unidade 2 do programa, como introdução ao tema “Estruturas de Repetição”.
<b>Pré - requisitos</b>	<ul style="list-style-type: none"><li>• Conhecimento do ambiente de trabalho da ferramenta Bricx Command Center V3.3.</li><li>• Noção de variável e atribuição de valores a uma variável.</li></ul>
<b>Objectivos</b>	<ul style="list-style-type: none"><li>• Compreender o funcionamento de uma estrutura de repetição num programa;</li><li>• Distinguir as estruturas de repetição;</li><li>• Definir condições simples.</li></ul>
<b>Organização</b>	<ul style="list-style-type: none"><li>• Trabalho em pares ou individual;</li><li>• Os alunos deverão experimentar as várias opções e comparar com outras soluções afim de discutirem os resultados.</li></ul>
<b>Material Necessário</b>	<ul style="list-style-type: none"><li>• Robot;</li><li>• Sensor de Luz;</li><li>• Computador;</li><li>• Bricx Command Center.</li></ul>

## Possível Solução dos Problemas

### Solução do Problema 1

```
program ar5;

uses unistd, dsound, BIGBLOKS;

var i: integer;

begin
  i:=1;
  while (i<=5) do
    begin
      avancaAC(1);
      recuaAC(1);
      i:=i+1;
    end;

    dsound_system(0);

  end.
```

### Solução do Problema 2

```
program repatepreto;

uses
  unistd, dsensor, BIGBLOKS;

var luz: integer;

begin
  ds_active(@SENSOR_2);
  repeat
    luz:=LIGHT(SENSOR_2);
    avancaAC(1);
  until luz<50;

  end.
```

